

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a
biomedicínského inženýrství

Digitální zpracování signálu na platformě Infinity
Project Technology Kit

Digital Signal Processing Based on Infinity Project
Technology Kit Platform

2012

Bc. Jiří Skalný

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání diplomové práce

Student: **Bc. Jiří Skalný**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2601T004 Měřicí a řídicí technika
Téma: Digitální zpracování signálu na platformě Infinity Project
Technology Kit
Digital Signal Processing Based on Infinity Project Technology
Kit Platform

Zásady pro vypracování:

1. Průzkum vlastností kitu, jeho otevřenosti/uzavřenosti vůči obecným systémům a jinému typu DSP, obvodové řešení, vlastností a funkce zákaznického obvodu pro komunikaci s prostředím LabVIEW formou rešerše.
2. Vytvoření LabVIEW aplikace pro DSP procesor (nastavení počtu zpracovávaných kanálů mono/stereo, implementace filtrů prvního a vyšších řádů pro horní/dolní propust, pásmovou propust/zádrž, implementace parametrického ekvalizéru s nastavitelnými parametry, implementace výhybky pomocí DSP pro dvoupásmovou aktivní reprosoustavu výhledově pro 4 pásmovou).
3. Vytvoření přehledného grafického prostředí jak pro konfigurace tak pro monitorování.
4. Ověření funkčnosti navržených filtrů pomocí bílého šumu.
5. Implementace funkce ABL (Adaptive Bass Linearization).
6. Reálné měření všech implementovaných filtrů (např. vstup bílý šum - výstup spektrální analyzátor), monitorování zpracovávaného signálu.
7. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] JAN, Jiří. *ČÍSLICOVÁ FILTRACE, ANALÝZA A RESTAURACE SIGNÁLŮ*. Brno: VUTUM, 2005. 427 s. ISBN 80-214-2911-9.
- [2] SHENOI, B.A. *INTRODUCTION TO DIGITAL SIGNAL PROCESSING AND FILTER DESIGN*. USA : Wiley & Sons, 2005. 440 s. ISBN 978-0-471-46482-2.
- [3] SMITH, Julius. *Stanford University* [online]. 2007 [cit. 2011-10-31]. INTRODUCTION TO DIGITAL FILTERS. Dostupné z : <<http://ccrma.stanford.edu/~jos/filters/filters.html>>.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jan Žídek, CSc.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Ing. Jiří Kozírek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení Studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě

4.5.2012

Jan Židka
.....
podpis studenta

Poděkování

Rád bych poděkoval panu doc. Ing. Janu Židkovi, CSc. za vedení mé diplomové práce, za jeho podněty, rady a připomínky. Taktéž bych rád poděkoval panu Ing. Robertu Hoškovi za konzultace a zajištění měřicího pracoviště ve firmě Bang & Olufsen.

Zvláštní poděkování bych chtěl věnovat mým rodičům za podporu během celého studia.

Abstrakt

Tato práce se zabývá návrhem digitálních filtrů, které jsou určeny pro použití v audio reproduktorových řetězcích. Pro digitální zpracování signálu bylo použito DSP, které bylo součástí vývojového kitu. Filtry byly navrženy pro dva kity, kterými byl kit NI Speedy – 33 a EVAL AD1940AZ.

NI Speedy – 33 je programovatelný prostřednictvím grafického vývojového prostředí LabVIEW, které má všestranné použití. Hlavním blokem, který byl v práci použit, je bikvadratický filtr. Ten realizuje jedinou přenosovou funkci dostupnou v LabVIEW DSP modulu. Pro návrh koeficientů bikvadratického přenosu byla použita bilineární transformace, jež využila Butterworthovy aproximace jako předlohy. Dalším použitým blokem je reálná FFT, která užívá pro filtraci frekvenční spektrum. Na kitu NI Speedy – 33 byla implementována funkce ABL, která využívá normu ISO 226 jako svou předlohu.

Kit EVAL AD1940AZ, jež je také programovatelný přes grafické vývojové prostředí s názvem SigmaStudio, je určen výhradně pro zpracování digitálního audio signálu.

Jednotlivé naměřené průběhy kitů jsou vzájemně porovnány a z těchto výsledků vyplývají jejich přednosti a použitelnost.

Klíčová slova: DSP, LabVIEW, Infinity Project, NI Speedy – 33, Butterworth, bikvadratický filtr, reproduktorová výhybka.

Abstract

This paper describes the design of digital filters that are designed for use in audio loudspeakers systems. DSP was used for digital signal processing which was the part of a development kit. Filters were designed for two kits, NI Speedy - 33 and EVAL AD1940AZ.

NI Speedy - 33 is programmable through a graphical development environment LabVIEW, that has universal use. The main block, which was used in the thesis, is the biquadratic filter. It realizes the only transfer function available in LabVIEW DSP Module. Bilinear transformation was used for the design of biquadratic transmission coefficients and employed Butterworth approximation as a pattern. Real FFT is another used block that uses a frequency spectrum for filtration. The function ABL was implemented on the kit NI Speedy – 33 that uses ISO 226.

Kit EVAL AD1940AZ, which is also programmable through graphical development environment, is designed exclusively for digital audio signal processing.

The individual courses of measured kits are compared with each other and these results suggest their advantages and applicability.

Key words: DSP, LabVIEW, Infinity Project, NI Speedy – 33, Butterworth, biquadratic filter, crossover.

Seznam použitých symbolů a zkratek

A/D	Analogově digitální převodník
ABL	Adaptive Bass Linearization
ALU	Arithmetic Logic Unit
ANSI	American National Standards Institute
CMOS	Complementary Metal–Oxide–Semiconductor
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
D/A	Digitálně analogový převodník
DC	Direct Current
DMA	Direct Memory Access
DP	Dolní propust
DSP	Digital Signal Processing
EPROM	Erasable Programmable Read-Only Memory
EQ	Equalizer
FDAT	Filter Design and Analysis Tool
FIFO	First In, First Out
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
GPIB	General Purpose Interface Bus
GUI	Graphical User Interface
HW	Hardware
I/O	Input/output
I2S	Integrated Interchip Sound
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IIR	Infinite Impulse Response
ISO	International Standard Organisation
LD	Ladder Diagram
LSB	Least Significant Bit
MAC	Multiply–accumulate
MFLOPS	M Floating-point Operations Per Second
MIPS	M Instruction Per Second
MLSSA	Maximum length sequence system analyser
MSB	Most Significant Bit
NRDF	Non-recursive Filter
PAL	Programmable Array Logic
PCI	Peripheral Component Interconnect
PDA	Personal Digital Assistant
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation

PXI	PCI Extensions for Instrumentation
RAM	Random Access Memory
RDF	Recursive digital filter
SNR	Signal-to-noise Ratio
SPL	Sound Pressure Level
SW	Software
TDM	Time Division Multiplexing
THD	Total Harmonic Distortion
UB	Universal Bus
UML	Unified Modeling Language
USB	Universal Serial Bus
VEE	Visual Engineering Environment
VI	Virtual Instrumentation

Obsah

1	Úvod.....	1
2	Vlastnosti kitu	3
2.1	Srovnání DSP procesorů	3
2.1.1	Architektura DSP	5
2.2	Kit NI Speedy – 33.....	6
2.3	Nástroje pro programování	8
2.3.1	Procedurální programování	9
2.3.2	Grafické programování	10
2.4	LabVIEW	12
2.4.1	LabVIEW DSP modul.....	16
3	Analýza obvodů v reproduktorových soustavách	18
3.1	Fyzikální vlastnosti reproduktorů	18
3.2	Typy filtrů	20
3.2.1	Analogové	21
3.2.2	Digitální	21
3.2.3	Srovnání číslicového a analogového zpracování signálů	25
3.3	Analogo – digitální a digitálně - analogová konverze	26
3.3.1	Vzorkování	27
3.3.2	Kvantování	28
3.3.3	Kódování	29
3.4	Aplikace filtrů pro DSP procesor	30
4	Návrh aplikace s GUI rozhraním	32
4.1	Filtrace v časové oblasti	32
4.1.1	Filtrace užívající expresní funkci	32
4.1.2	IIR filtr 2. Řádu	34
4.2	Filtrace ve frekvenční oblasti	49
5	Ověření funkčnosti navržených filtrů.....	54
5.1	Ověření dolní, horní a pásmové propusti	54
5.2	Ověření pásmové zádrže	55
5.3	Ověření parametrického ekvalizéru	56
5.4	Ověření funkce ABL	57
6	Implementace funkce ABL	58
6.1	Simulace funkce ABL v prostředí MATLAB	60
6.2	Návrh algoritmu funkce ABL	62
6.3	Hlavní panel funkce ABL	64
7	Kit EVAL AD1940AZ.....	65
7.1	Úlohy realizované na AD1940.....	65
8	Reálná měření	69
8.1	Měření úloh na kitu NI Speedy - 33	69
8.2	Měření úloh na kitu EVAL AD1940AZ	76
9	Závěr	80
10	Použité zdroje.....	82
11	Přílohy.....	84

1 Úvod

Elektronika patří k nejdynamičtější se rozvíjejícím odvětvím. Prudký vývoj elektroniky můžeme datovat již do 40. až 50. let 20. století, kdy se začaly užívat polovodičové, tehdy ještě germaniové diody. Později to byly tranzistory germaniové, křemíkové a první integrované obvody. Před padesáti lety by člověk jistě nevěřil, že za několik desítek let bude technologie natolik rozvinutá, že se do jedné tenké destičky vejde hned několik desítek milionů tranzistorů. Touto součástí je chip, na němž může být realizován i celý procesor.

Procesorem neboli CPU se rozumí součástka, která vykonává strojový kód spuštěného programu, který je složen z jednotlivých strojových instrukcí, jež jsou uloženy v operační paměti počítače. Procesor ke své činnosti potřebuje paměť, vstupní a výstupní zařízení. To vše může být umístěno na jedné desce plošných spojů. Pokud integrujeme do jednoho chipu jak ALU, tak i radič, dostáváme tzv. mikroprocesor. První mikroprocesor i4004 byl 4 bitový a firma Intel jej uvedla na trh v roce 1971. Tento mikroprocesor našel využití především v kalkulátorech. Postupně byly vyvíjeny další mikroprocesory, které reagovaly na poptávku trhu. S příchodem digitálního zpracování signálů vyvstala potřeba speciálních mikroprocesorů, které umožňovaly rychlý výpočet algoritmů používaných při zpracování těchto signálů.

Vznik prvního digitálního signálového procesoru neboli DSP můžeme datovat na přelom 70. a 80. let minulého století. Roku 1979 byl uveden na trh signálový procesor I2920 firmy Intel. DSP I2920 nebyl mezi konstruktéry příliš oblíben kvůli jeho omezeným možnostem, avšak poukázal na mezeru v technických prostředcích. Tento fakt vedl mnoho firem k vývoji signálových procesorů, jejichž architektura byla přizpůsobena pro digitální zpracování signálů. Velmi brzy se DSP staly nejrychlejšími mikroprocesory na světě. Toto postavení ztratily s masovým rozvojem procesorů určených pro osobní počítače. [1]

Od této doby uplynulo bezmála 30 let a signálové procesory byly natolik zdokonaleny, že plně výkonnostně vyhoví i v náročných aplikacích. Nyní nachází uplatnění nejen při zpracování signálů audiovizuálních či telekomunikačních, ale také ve stavebnictví, medicíně, strojním či chemickém průmyslu.

Tato práce se zabývá filtrací audio signálu, která je potřebná pro dělení frekvenčních pásem v reproduktorových soustavách. V běžných reproduktorových soustavách se používá analogové dělení pásem pomocí kondenzátorů a indukčností. Použití digitální filtrace přináší přesné dělení kmitočtů, možnost kompenzace fázových posunů, snadnou změnu parametrů či použití adaptivních filtrů. Pro digitální filtraci byl použit digitální signálový procesor TMS320VC33 firmy Texas Instruments. Procesor je osazen ve vývojovém kitu NI Speedy - 33 firmy National Instruments. Kit je programován pomocí grafického vývojového prostředí LabVIEW, které je zahrnuto v programu Infinity Project. Hlavním úkolem projektu je demonstrace možností digitálního zpracování signálu studentům technických oborů. Procesor, kit i vývojové prostředí jsou blíže popsány v následující kapitole.

Třetí kapitola popisuje obvody reproduktorových soustav, srovnává analogové a digitální filtry. V kapitole je dále vysvětlen návrh digitálních IIR filtrů s užitím Butterworthovy aproximace. V navazující kapitole je popis algoritmů jednotlivých úloh (dolní, horní a pásmové propusti, pásmové zadržky a parametrického ekvalizéru). Ke všem úlohám bylo vytvořeno přehledné grafické ovládání, kterým lze měnit nastavení filtrů.

Pátá kapitola se zabývá ověřením funkčnosti navržených filtrů. Pro simulaci vstupního signálu byl použit programový blok Simulate Signal, který je součástí LabVIEW. Celá simulace probíhala pouze na HW kitu.

V šesté kapitole je popsána funkce ABL, která se zabývá fyziologickou regulací hlasitosti v oblasti basových kmitočtů. Také je zde popsán algoritmus funkce a grafické rozhraní.

Pro srovnání kitu NI Speedy – 33 byly v sedmé kapitole některé úlohy (horní, dolní a pásmová propust, parametrický ekvalizér a funkci ABL) naprogramovány na kitu EVAL AD1940AZ. Kit EVAL AD1940AZ je vyráběn firmou Analog Devices.

V osmé kapitole jsou popsány výsledky měření navržených filtrů. Pro měření byl použit audio analyzátor ROHDE & SCHWARZ UPV.

Závěrečná kapitola je věnována zhodnocení výsledků a srovnání obou kitů.

2 Vlastnosti kitu

Tato kapitola bude věnována především hardwaru a softwaru kitu NI – Speedy 33 a popisu vyplývajících vlastností. V první části jsou popsány DSP procesory a srovnání vybraných zástupců.

2.1 Srovnání DSP procesorů

Od doby, kdy firma Texas Instruments v roce 1982 dala na trh první komerčně úspěšný procesor TMS 32010, byla vyvinuta a vyrobena nespočetná řada dalších procesorů. Hlavními výrobci se staly firmy Texas Instrument, Lucent, Analog devices a Motorola, později dceřiná společnost Freescale. Jejich srovnání není jednoduché, protože jejich produkty jsou často určeny pro rozdílné aplikace, pro které jsou důležité jiné parametry, a tedy i jiná kritéria pro srovnání.

Procesory můžeme rozdělit do tří kategorií podle výkonu, ceny a kategorie využití. [4][5][24]

Low Cost – levná třída

V této třídě je pravděpodobně vyráběno největší množství DSP procesorů. Procesory v této kategorii pracují s šířkou slova 16 bitů a obvykle nepodporují operace v plovoucí řádové čárce. Procesor obsahuje jednu násobičku, která zvládá násobení v jednom strojovém cyklu nebo může obsahovat MAC jednotku, která umí násobit dvě čísla a výsledek přičíst do akumulátoru. Samozřejmě obsahuje datové sběrnice – ty jsou obvykle dvě a mají šířku 16 bitů a sběrnice adresní. Pracovní kmitočet bývá 20-50 MHz. Tyto DSP mají dostatečný výpočetní výkon pro běžné aplikace při malé spotřebě elektrické energie. Typickými zástupci v této třídě jsou například: Analog Devices ADSP-21xx, Texas Instruments TMS320XC2xx, Motorola DSP560xx, Lucent DSP16xx.

Procesory v této třídě nacházejí uplatnění např. v digitálních telefonech, harddiscích a v celé řadě dalších aplikací. [4][5][26][28][29]

Midrange – střední třída

Procesory této třídy dosahují vyšších výkonů oproti předchozí kategorii. Mají vyšší taktovací kmitočet, který se pohybuje v rozmezí 100-150MHz. Obvykle pracují s šířkou slova 32 bitů a umožňují práci s instrukcemi v plovoucí řádové čárce. Může být zvýšen počet datových i adresních sběrnic, DMA kanálů pro přímý přístup do paměti. Zpravidla obsahují rychlé cache paměti a případně i další doplňkový hardware (seriové porty, paralelní porty, časovače a další). Typickými zástupci v této třídě jsou například: Texas Instruments TMS320C54xx, kam patří i TMS320VC33, který je použit v kitu NI Speedy -33, zástupcem firmy Motorola je DSP563xx, dále procesory SHARC ADSP 2106x a ADSP 2116x, které uvedla na trh uvedla firma Analog Device.

Procesory této třídy jsou typicky osazovány do bezdrátových telekomunikačních aplikací a do vysokorychlostních modemů. Mají dostatečný výkon a přijatelnou spotřebu elektrické energie. [28][29][4][5][25]

High End - vyšší třída

V této třídě se již nedosahuje zvýšení výkonu prostým zvyšováním kmitočtu anebo rozšířením datových sběrnic na 64bitů či zdvojením ALU nebo násobiček. Začínají se zde uplatňovat jiné typy architektur např. VLIW (very long instruction word). Procesor obsahuje větší počet paralelních výpočetních jednotek. V případě procesoru TMS320C62xx je takových jednotek 8. Každá jednotka je ovládána jinou částí instrukce VLIW. Představiteli této kategorie jsou procesory TMS320C32xx a TigerSHARC od firmy Analog Devices. Procesory se uplatňují v nejnáročnějších aplikacích. [4][5][28][29]

Pro bližší srovnání (viz obr. 1) jsou zde uvedeny tři DSP procesory, které byly vyráběny přibližně ve stejné době. Jsou to Texas Instruments TMS320VC33, Analog Devices Sharc ADSP21020 a Motorola DSP56301. [4][5][28][29]

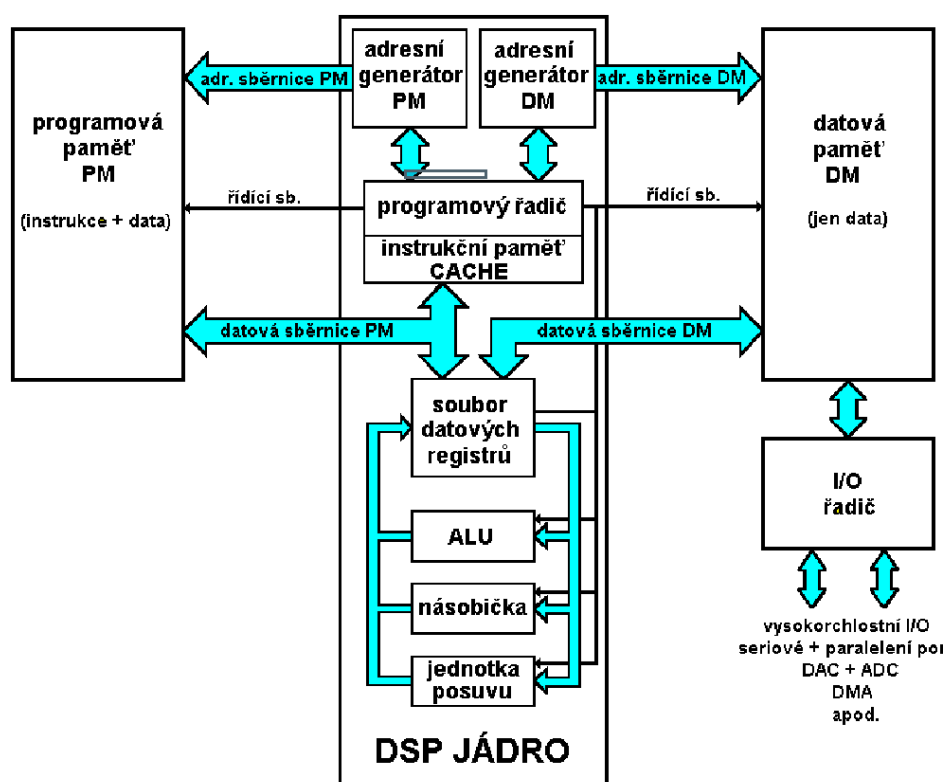
	TMS320VC33	ADSP2120	DSP56301
instrukční cyklus	13ns	30ns	15ns/12,5ns; dle teploty
instrukce násobení	1cykl;13ns	1cykl;30ns	1cykl;15ns
instrukce dělení	36cyklu,468ns	6cyklu,180ns	24cykly; přesnost 24bit
kmitočet MHz	77	33	66/80; dle teploty
MIPS	75	33.3	66/80; dle teploty
MFLOPS	75	66	nemá FPP
Fixed point data format	16/32bit	16/32bit	16/24/48
Floating point data format	32/40	32/40	nemá FPP
architektura	harvardská	harvardská	harvardská
MAC (násobič)	ano	ano	ano
barrel shifter (posouvač)	32bit	32bit	56bit
operandy v instrukci	2-3	2-3	2-3
interní datové sběrnice	4x32 bit	40,48bit	5x24bit
interní adresní sběrnice	3x24bit	24,32bit	4x24bit
interní paměti	1k x 32bit		4k x 24bit; program
	1k x 32bit		2k x 24bit; data
	16k x 32bit		2k x 24bit; data
	16k x 32bit		
paměť cache	64 x 32bit	32 x 48bit	dle konfigurace
boot loader	ano	ne	ano
adresní generator	2x24bit	32bit+24bit	2x24bit
kanály DMA	1	nemá int. paměť	6
seriový asynchronní port	1	0	1
seriový synchronní port	0	0	2
externí sběrnice	A24, D32	A24,A32,D48,D32	A24,D24,PCI/GPIB/UB 52bit
timer	2	1	3
externí přerušení	4	4	4
napájení	1,8V; 3,3V	4,5-5,5V	3-3,6V
spotřeba	200mW	1600mW	420mW
Poznámka: DSP56301 umožňuje práci s dvojnásobnou délkou slova a jeho 52bitová externí sběrnice může být nakonfigurována na PCI, GPIB nebo UB(Universal Bus)			

obr. 1. Srovnání parametrů vybraných DSP

[4][5][24][25][26][8][9]

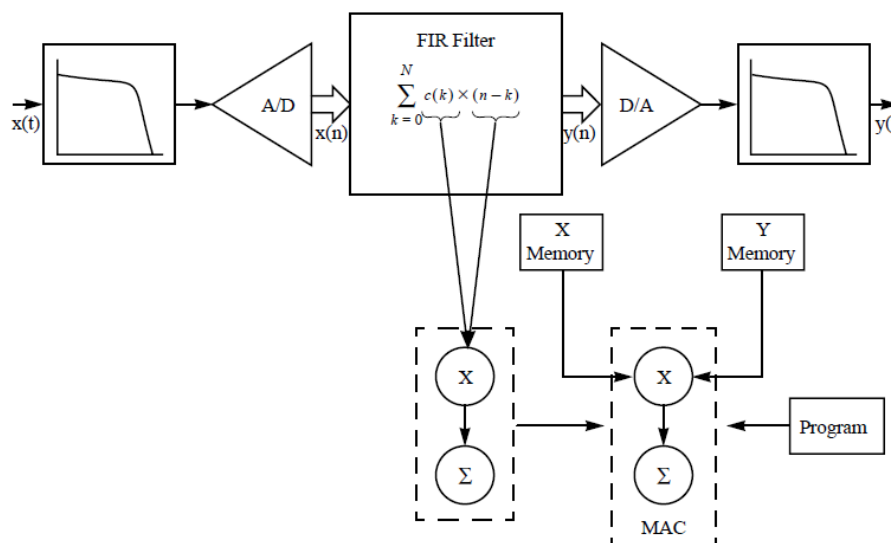
2.1.1 Architektura DSP

Na obrázku je znázorněno typické schéma DSP procesoru s harvardskou architekturou. Rozdíl proti procesoru s von Neumannovou architekturou je především v použití dvou pamětí. Jedna paměť slouží k uložení dat a druhá pak k uložení dat a instrukcí. Každá paměť má vlastní adresní i datovou sběrnici. Tak lze v jedné instrukci adresovat dva operandy a přenést je ze dvou pamětí do ALU. Další operand může být umístěn v registrech procesoru. Procesor s von Neumannovou koncepcí má data i instrukce umístěny v jedné paměti a je třeba je postupně, sekvenčně vyčítat a přenášet do ALU. [4][5]



obr. 2. Harvardská architektura

V horní části obrázku jsou nakresleny dva adresní generátory, které generují adresy pro adresování pamětí. Generátory adres pracují paralelně s ALU. Tímto mechanismem se zrychlí přístup k pamětem. Častou operací při digitálním zpracování signálu je násobení dvou operandů a přičítání výsledku do registru. [4][5][1][8][9]



obr. 3. Výpočet FIR filtru v procesoru s harvardskou architekturou. [27]

DSP procesor je vybaven MAC, která vynásobí dva operandy a výsledek přičte do registru v jednom cyklu. V tomto cyklu generátory adres adresují následující operandy. Jednotka MAC je nezávislá na ALU. To, co DSP procesor s harvardskou architekturou vykoná v jednom cyklu, by procesor se starší von Neumannovou architekturou vykonával až v 5 cyklech. [4][5][1]

2.2 Kit NI Speedy – 33

Tento kit je konkrétní aplikací procesoru Tms320vc33. Jeho všechny možnosti nejsou na desce plošného spoje uplatněny, a tak poskytuje omezené možnosti využití. Možnosti kitu vyplývají z níže popsaných konektorů na obr. 4. NI Speedy – 33 byl vytvořen pro studijní účely v programu Infinity Project a pro demonstraci standardních příkladů použití tak plně dostačuje.

Celý kit NI Speedy-33 je realizován na jedné desce plošného spoje o rozměrech 98x127mm. Deska plošného spoje je umístěna do robustní hliníkové krabičky, která zajišťuje dobrou mechanickou ochranu.

Deska je osazena několika elektronickými obvody, z nichž nejdůležitějším je již zmíněný DSP procesor. Ten však ke své práci nutně potřebuje další obvody.

Jedním je PCM 3003E, jehož výrobcem je Texas Instrument. PCM3003E je kvalitní A/D a D/A převodník s vzorkovací frekvencí 48kHz a 20 bitovou hloubkou s THD -86dB. Analogovo – digitální převod disponuje 90dB SNR a odstup signálu od šumu při využití D/A nabývá hodnoty 94dB. Zejména tento obvod udává kvalitu zpracovávaného signálu. Z uvedených parametrů je zřejmé, že by mohl dostačovat k realizaci úloh v zadání diplomové práce.

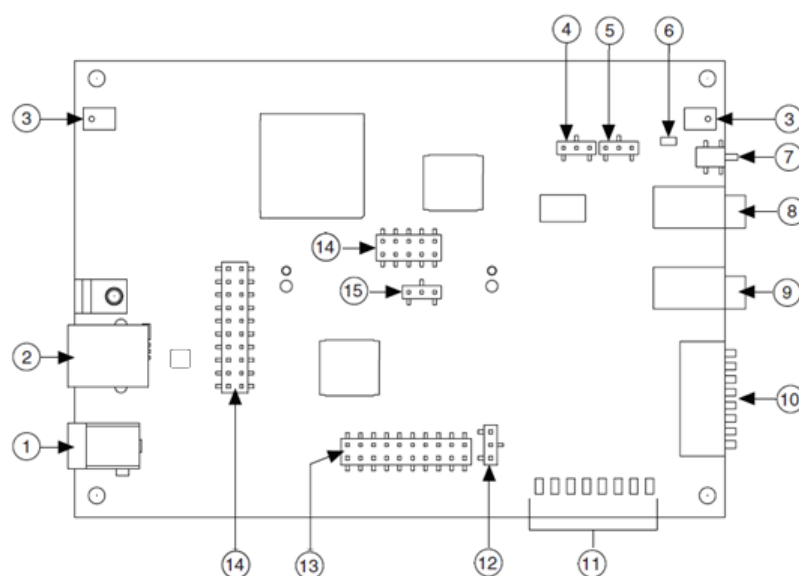
Dalším, nemálo důležitým obvodem je CY37064VP100-100AXI firmy Cypress. Jedná se o součástku CPLD (complex programmable logic device), což je jednoduché hradlové pole skládající se z buněk PAL. Přesná funkce obvodu není známa, pravděpodobně však bude zprostředkovávat komunikaci mezi PC a DSP.

Obvod pod označením FT245BL má však funkci jasnou. V tomto případě představuje dvousměrný konvertor se zásobníkem FIFO, kdy jeden směr je reprezentován sériovou USB sběrnici a druhý paralelní.

Paměti kitu jsou realizovány obvody AT49BV040B a AT93C46w od firmy Atmel, které kitu poskytují kapacitu 8x512 kbyte + 8x128 bit. V prvním případě se jedná o rychlou flash paměť založenou na CMOS technologii, v druhém je to pak paměť EPROM.

Posledním obvodem, o kterém se dle označení dalo něco zjistit je stabilizátor 7805. Jeho funkcí je stabilizovat napětí připojeného externího zdroje.

Bližší popis kitu není možný, protože výrobce neuveřejnil schéma zapojení. [38][24]



obr. 4. Konektory kitu

1. Napájecí konektor – slouží pro napájení modulu z externího zdroje a umožní tak jeho použití bez nutnosti připojení PC.
2. USB konektor - zajišťuje komunikaci mezi NI Speedy 33 a uživatelským PC s prostředím LabVIEW. Jestliže je modul připojen přes USB, není nutné použití externího zdroje.
3. Integrované mikrofony.
4. Propojka pro levý kanál – umožňuje přepínání mezi integrovaným mikrofonom a linkovým vstupem levého kanálu.
5. Propojka pro pravý kanál.
6. Led indikace zapnutí.
7. Reset – hardwarové tlačítko pro reset modulu.
8. Audio stereo vstup.
9. Audio stereo výstup.
10. Digitální vstup – je realizován pomocí osmi přepínačů.
11. Digitální výstup – je realizován pomocí osmi led diod.

12. Flash boot jumper – slouží pro možnost zavedení programu z paměti.
13. Rozšiřující digitální I/O konektor – jedná se o 20ti pinový konektor, na kterém najdeme napájení, zem, 8 digitálních vstupů a 8 digitálních výstupů.
14. Analogový I/O konektor – jedná se o konektor, na kterém je vyveden analogový vstup a výstup a umožněno tak snadnější připojení případného modulu.
15. Potvrzení zápisu flash – tento konektor určuje povolení zápisu do flash paměti. Je určen k ochraně zapsaného programu v paměti.

2.3 Nástroje pro programování

Nástroje pro programování jsou zpravidla vývojová prostředí (IDE) nebo překladače (compiler), které usnadňují vývojářům softwaru implementaci zvolených algoritmů. Vývojovým prostředím se zde myslí program realizující prostředí, které programátorovi umožní vytvoření vlastního programu. Vývojová prostředí jsou navržena tak, aby maximálně zproduktivnila práci programátora. Zpravidla je IDE orientováno na jeden konkrétní programovací jazyk. Jak již bylo zmíněno v podkapitole 2.1, existuje více výrobců DSP procesorů, kde každý výrobce nabízí i odpovídající vývojové prostředí. Pro DSP procesory Analog Devices je to VisualDSP++, který používá k programování jazyky C++, C a assembler. Dalším velkým výrobcem DSP je Texas Instrument, který nabízí vývojové prostředí Code Composer Studio užívající jazyka C a C++. Posledním velkým zástupcem ve výrobě DSP je Motorola/Freescale, který k programování užívá IDE CodeWarrior s jazykem C/C++ a assembleru.

Programování je proces, který se věnuje návrhu algoritmu, jeho implementaci s následným testováním tohoto algoritmu. Programovací jazyk je prostředek pro zápis algoritmů, které mohou být implementovány na zvolené platformě. Zápis algoritmu prostředky zvoleného programovacího jazyka se nazývá program.

Programovací jazyky se dělí podle několika hledisek. Podle míry abstrakce (na vyšší a nižší programovací jazyky) a dle způsobu překladů a spuštění (na kompilované či interpretované).

U kompilovaných jazyků se kód programu před spuštěním nejdříve přeloží do (číslcového) strojového kódu. Kompilátor je tedy překladač, který převede vyšší programovací jazyk na nižší. Výsledkem je vyšší rychlost, ale i větší náročnost na správně zapsaný program. Interpretované jazyky jsou ty, které ke své práci potřebují tzv. interpreter. To je počítačový program, který umožní přímo vykonávat zápis jiného programu v jeho zdrojovém kódu a ve zvoleném programovacím jazyce. Výsledná aplikace spuštěná pomocí interpretu bude pomalejší, „překlad“ se bude provádět stále při běhu programu. Tento způsob je však jednodušší na naprogramování a tím je ho zpravidla možné rychleji odladit a uvést do provozu a lze jej i snadněji testovat. Další výhodou je jednoduchá přenositelnost mezi různými platformami (Windows, Linux).

Mezi nižší programovací jazyky patří např. assembler, což je překladač do strojového kódu, který je srozumitelný procesoru. Assembler je uživatelsky nepřívětivý, jelikož nenese žádnou abstrakci kódu. Z tohoto důvodu se nižší programovací jazyky pro programování aplikací běžně nepoužívají. Množina vyšších programovacích jazyků je značně rozsáhlejší a můžeme ji rozdělit, např. podle metody zápisu algoritmu, na textové a grafické. Textové jazyky mohou být procedurální, kam patří objektově orientované a strukturované programování. [37][36][15][19]

2.3.1 Procedurální programování

Procedurální programování je metoda zabývající se formulací řešených problémů v programovacím jazyce. Tuto metodu můžeme rozdělit z hlediska návrhu softwaru na metodu „shora dolů“, která by byla zastoupena strukturovaným programováním a metodou „zdola nahoru“, které by odpovídal objektově orientovaný přístup.

Strukturované programování je technika, kdy se implementovaný algoritmus rozděluje na dílčí úlohy, které se spojí v jeden celek. U této metody je dán důraz především na srozumitelnost, čitelnost a bezpečnost programu. K implementaci se proto využívá vybraných řídicích struktur, ostatní nejsou povoleny např. skoky.

Objektově orientovaný přístup zpočátku definuje jednotlivé stavební buňky programu „objekty“, z nichž je celý program poskládán. Tento přístup napomáhá jednoduchému opětovnému užití již vytvořeného objektu. Každý objekt je instancí třídy, ze které byl vytvořen. Třída definuje množinu činností, které mají objekty dané třídy společné. Jednotlivé objekty stejné třídy se od sebe liší určitými vlastnostmi - „atributy objektů“. Činnosti, které objekty mohou vykonávat, se nazývají „metody“.

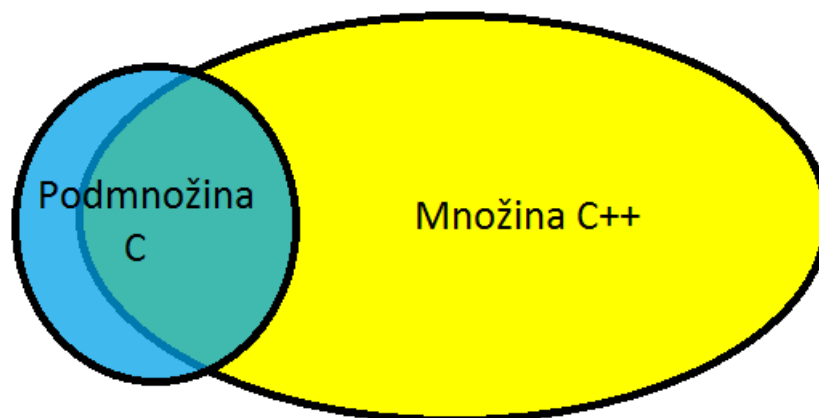
Z předchozího textu podkapitoly je zřejmé, že velmi častým zástupcem jazyků, které se užívají pro programování DSP, jsou jazyky C a C++, proto zde budou stručně popsány.

První zmínky o jazyku C můžeme najít v roce 1978, kdy Brian W. Kernighan a Denis M. Ritchie vydávají knihu „The C Programming Language“. Tento standard se po svých tvůrcích nazýval K&R. Z tohoto standardu vychází o deset let mladší ANSI C, jehož součástí je i specifikace použití funkcí z knihoven souborů (.H). Dnes je aktuální standard ISO/IEC 9899:1999, který má oproti předchůdci nové prvky.

Jazyk C je nízkourovňový strukturovaný programovací jazyk, který má úsporné vyjadřování, avšak podporuje mnoho operátorů a moderní datové typy. Pomocí jazyka C byl napsán téměř celý systém UNIX. Je tedy zřejmé, že pro určité aplikace bude efektivní.

Jazyk C++ vychází z původního jazyka C. Hlavním rozdílem je rozšíření o možnost objektového přístupu. O tuto změnu se přičinil Bjarne Stroustrup, který v roce 1986 publikoval knihu „C++ Programming Language“. Důvodem vydání knihy byly negativní zkušenosti s tehdejšími moderními jazyky Simula. Program v něm bylo sice možné efektivně naprogramovat, avšak v reálném sběru dat se ukázalo, že výpočty jsou příliš pomalé. Naopak nižší jazyk BCPL (předchůdce C) prokázal dostatečnou rychlost výpočtu na úkor programovacích nároků.

V průběhu 90. let byl jazyk C++ stejně jako předchůdce normalizován. Zajímavé je, že ISO norma C++ ovlivnila ISO normu jazyka C, což se zpětně projevilo na další ISO C++. Tento vývoj obou jazyků zabezpečil mezi sebou určitou kompatibilitu. Lze tvrdit, že jazyk C je podmnožinou jazyka C++, avšak má navíc několik funkcí (viz obr. [37][36][15][19]



obr. 5. Grafické srovnání jazyka C a C++

Z obrázku je zřejmé, že kód, který by spadl do množiny C++ a C, bude možno přeložit kompilátorem určeným jak pro C++, tak i pro C. Zajímavostí je, že první kompilátor jazyka C++ jazyk nejdříve přeložil do jazyka C, pak teprve do strojového kódu. Jelikož jazyk podporuje jak strukturovaně psaný kód, tak i objektový přístup, jedná se tak o určitý přechod mezi těmito možnostmi programování. Program vytvořený pomocí jazyka C++ může tedy obsahovat obě metody přístupu. [37][36][15][19]

2.3.2 Grafické programování

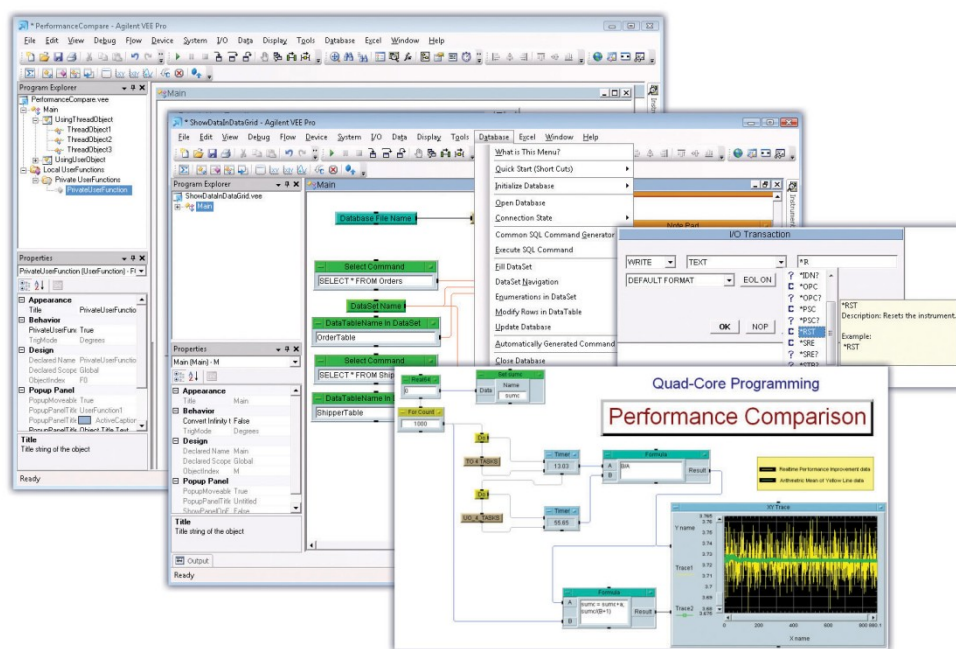
Grafické programování, někdy nazývané jako vizuální programování, se v oblasti výroby, automatizace a přístrojové techniky využívá již přes 20 let. Vývojová prostředí, která užívají metod grafického programování, se od klasických textových programovacích jazyků značně liší. Vizuální programování využívá grafickou nabídku prvků, se kterou programátor vytváří, uspořádává a propojuje ikony představující různé programátorské a datové struktury či funkce. Výhodou grafického programování je vyšší abstrakce, intuitivnost a přehlednost, která programátorům pomůže k rychlému pochopení samotného vývojového prostředí nebo kódu programu. Dá se říci, že tato metoda by se dala přirovnat k UML diagramu datových toků, kdy datový tok představuje spojnici („drát“) mezi jednotlivými funkčními bloky.

Po dokončení grafického znázornění daného problému do podoby blokového diagramu lze takto vytvořený blokový diagram zkompilevat do standardního spustitelného programu a následně jej přímo spustit. [11][15][39][37][36]

Jako příklady užití vizuálního programování zde uvedu několik zástupců, které tuto metodu ve svém konceptu programování užívají.

Agilent VEE (Visual Engineering Environment) je produkt společnosti Agilent Technologies, která se primárně zaměřuje na výrobu elektronických přístrojů jako jsou osciloskopy, voltmetry, generátory, logické a spektrální analyzátory. Agilent VEE je grafické programovací prostředí, které se užívá pro návrh automatizovaného měření, analýzu a ukládání dat. První verze tohoto prostředí byla vydána 9.5.1991 pod názvem HP VEE s označením A.00.00, která byla určena jako vizuální prostředí pro přístroje firmy Hewlett Packard. Později

bylo prostředí přejmenováno na již zmíněné Agilent VEE, které je orientováno převážně na přístroje stejnojmenné firmy. Tento fakt je jistým omezením a zamezuje tak univerzální použitelnosti. Prostředí však disponuje možností použití technologie ActiveX, čímž umožní použití jednotlivých komponent běžících pod operačními systémy Windows, které mohou být naprogramovány např. jazykem C/C++ a dalšími jazyky kompatibilními s platformou .NET. Tím je zpřístupněno např. zasílání emailových zpráv či ukládání dat do MS Excelu. Další předností prostředí Agilent VEE je komunikace se Signal Processing Toolboxem z prostředí MATLABu. Je tak určitou konkurencí propracovanějšího IDE LabVIEW. Příklad zobrazení je na obrázku obr. 6. [11][15][39][37][36]

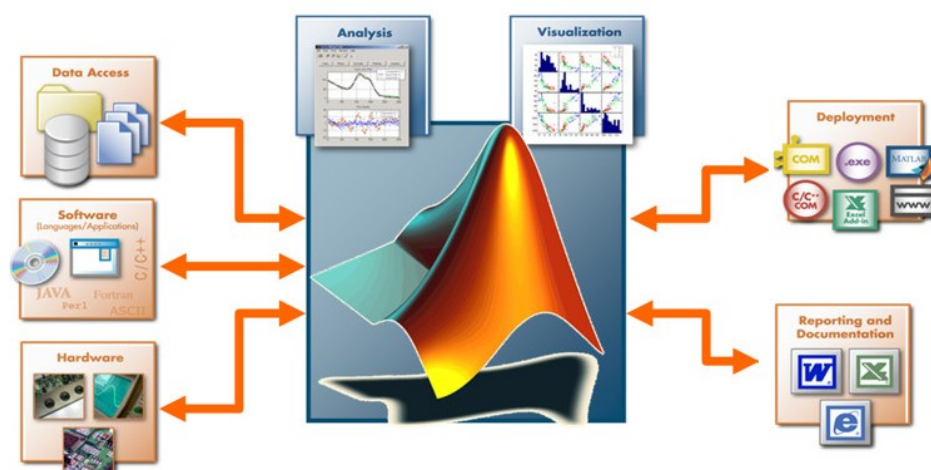


obr. 6. Prostředí Agilent VEE

MATLAB (Matrix Laboratory) je integrované prostředí pro vědeckotechnické výpočty, modelování, simulace, analýzu a prezentaci dat, návrhy algoritmů, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. Tyto vlastnosti vystihuje obr. 7.

První verzi programu vydala firma MathWorks roku 1984 ve verzi 1.0. Od této doby se stal MATLAB celosvětovým standardem v oblasti technických výpočtů, simulací a je používán v různých profesích. Ke své práci používá řadu funkcí a výkonný skriptovací jazyk 4. generace. Důležitou nadstavbou MATLABu je Simulink, který je určen pro modelování a simulaci dynamických systémů, který využívá algoritmy MATLABu pro numerické řešení diferenciálních rovnic. Simulink byl poprvé uveden s MATLABem verze 4. v roce 1992. Jeho hlavní předností je užívání grafických prvků pro modelování řešených úloh, díky nimž se práce s jeho využitím stává intuitivní a má stejné výhody jako další zde uvedené příklady. MATLAB taktéž umožňuje použití ActiveX prvků, použití platformy .NET nebo použití Java knihoven. Nechybí ani podpora vybraného hardwaru, a tak je možné využít prostředí pro jejich naprogramování. Tato metoda spočívá ve vytvoření diagramu, který je následně přeložen např.

do jazyka C. Tento vygenerovaný kód je pak nahrán přímo do paměti HW. Tato práce se zabývá filtrací signálů, jejichž funkce má ve svém portfolio i prostředí MATLAB, který má k dispozici několik DSP knihoven, díky nimž je možná jednoduchá implementace vytvořených kódů filtrů pomocí Signal Processing Toolboxu. [11][15][39][37][36][19]



obr. 7. Grafické znázornění možností MATLABU

Dalším prostředím, které je založeno na vizuálním programování je LabVIEW. Ze zde srovnávaných IDE je nejpodobnější prostředí Agilent VEE. Má obdobné vlastnosti, avšak LabVIEW oproti Agilent VEE je rozsáhlejší a můžeme jej nazvat univerzální platformou. Protože kit NI Speedy – 33 je programován právě přes prostředí LabVIEW, bude mu věnována následující podkapitola.

Poslední příklad, který zde bude uveden, se týká LD diagramů. Pod pojmem LD se skrývají dvě anglická slova „Ladder Diagram“ (v překladu žebříkový diagram). Jedná se o grafický jazyk, určený pro programování automatických systémů, které jsou známy např. u PLC. LD. Je užíván již od II. světové války a je tak jedním z nejstarších, ale stále populárních jazyků, které jsou určeny k programování automatických systémů. Oproti předchozím příkladům se liší zejména dvěma hlavními faktory. Nejedná se o vývojové prostředí a pracuje s logikou, která má analogii v relových obvodech. Při spojení dvou bloků tak nebudeme vytvářet datový tok, ale přenos logické hodnoty. Dílčí logické funkce jsou vytvářeny jako sérioparalelní kombinace kontaktů, které spolu s výstupními funkcemi cívek relé, paměťovými funkcemi, čítači a časovači a dalšími funkčními bloky tvoří pomyslný žebřík, z čehož plyne název tohoto jazyka. [39][37][36] [11][15]

2.4 LabVIEW

Pod slovem LabVIEW se skrývá zkratka Laboratory Virtual Instrumentation Engineering Workbench a je tak označováno vývojové prostředí, které je dílem firmy National Instruments. Firma vznikla roku 1976 na Texaské univerzitě v Austinu. Jejími zakladateli byli James Truchard, Bill Nowlin a Jeff Kodovsky, který je považován za „otce“ LabVIEW. Hlavní myšlenkou zakladatelů bylo revoluční odstartování měření za pomoci počítače. Roku 1983 byla

firmou vyvinuta první počítačová deska s rozhraním GPIB (1972 – Hewlett Packard). To odstartovalo vývoj grafického programovacího prostředí LabVIEW. Již roku 1986 byla uvedena první verze LabVIEW pro Macintosh. Následující rok byla vytvořena verze LabWindows, která pracovala pod operačním systémem DOS.

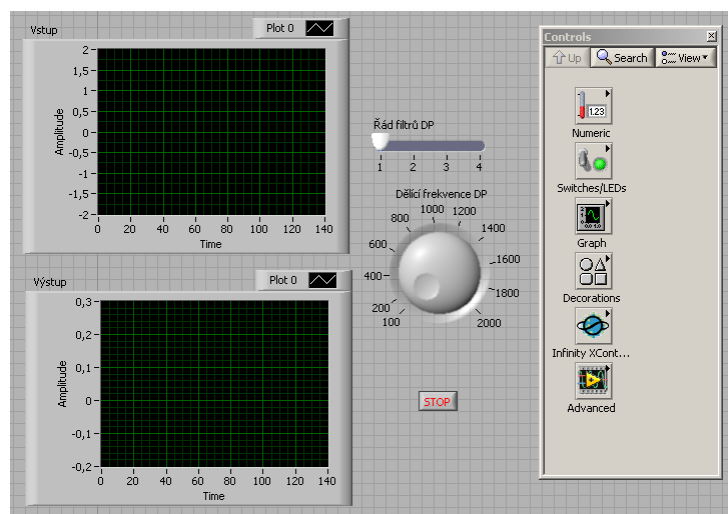
Prostředí užívá takzvaný „G“ jazyk (Graphical Language), který je srovnatelný s úrovní jazyka C/C++ a používá datových toků mezi jednotlivými bloky. Hlavní myšlenkou tohoto jazyka je zpřístupnit programování technikům, kteří však nejsou programátory. Hlavním potenciálem LabVIEW je široká podpora hardwaru a softwaru, která vytváří z tohoto prostředí platformu. Příklad výčtu vlastností, které jsou spjaté s HW a SW:

- DAQ
- GPIB
- Seriólový port
- PXI
- VXI
- Internetové připojení
- Sítě LAN
- ActiveX
- DLL a CIN
- Math script

Z předchozího výčtu je zřejmé, že virtuální instrumentace (odtud VI) dosáhla v prostředí LabVIEW vysoké integrace a užívá se ve všech třech fázích měření: sběr, analýza a zpracování dat. Díky těmto aspektům lze prostředí použít nejen pro programování virtuálních přístrojů s měřicími kartami, ale také k vytvoření programu pro DSP, FPGA nebo PDA. Vývoj prostředí nadále pokračuje a snaží se tak softwarově využívat plně narůstajícího výkonu HW. Například s příchodem vícejádrových procesorů, které jsou dnes běžně užívány, byly vytvořeny paralelní struktury a zpracování, jež podporují práci s více jádrovými procesory.

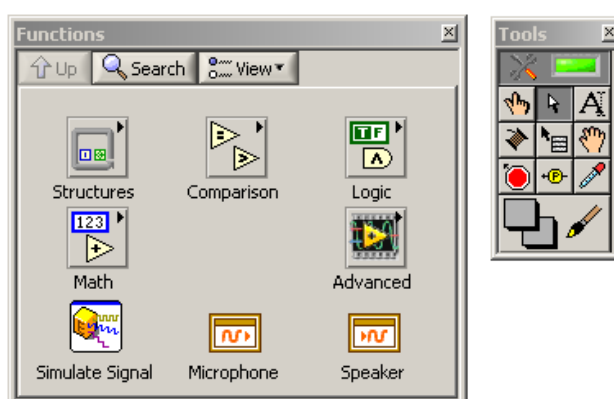
Návrh a realizace virtuálního přístroje ve vývojovém prostředí LabVIEW je intuitivní a vede tak ke snadnému pochopení jazyka. Prostředí je členěno do dvou hlavních oken.

První okno představuje čelní panel přístroje, který je složen ze dvou prvků, jež můžeme rozdělit na ovládací a indikační. Ovládací prvky představují řízení virtuálního přístroje uživatelem a jsou zastoupeny například: potenciometry, přepínači, posuvníky či tlačítka. Druhou skupinou jsou prvky indikační, které obsahují grafy, ledky nebo displeje s výstupní hodnotou. Použité prvky jsou dostupné z paletové nabídky ovládacích a indikačních prvků a mají obdobnou grafiku, jakou můžeme spatřit u reálných přístrojů. Pomocí těchto grafických prvků se vytváří grafické uživatelské rozhraní GUI. Při vložení ovládacího nebo indikačního prvku do VI se automaticky generuje příslušný blok v druhém okně – blokovém diagramu. [11][15][39]



obr. 8. Příklad zobrazení hlavního panelu s nabídkou kontrol

V okně blokového diagramu je vytvářen vlastní kód aplikace. Uživatel má ke své práci k dispozici dvě palety - obr. 9.



obr. 9. Zobrazení palet při tvorbě blokového diagramu v LabVIEW pro Infinity projekt

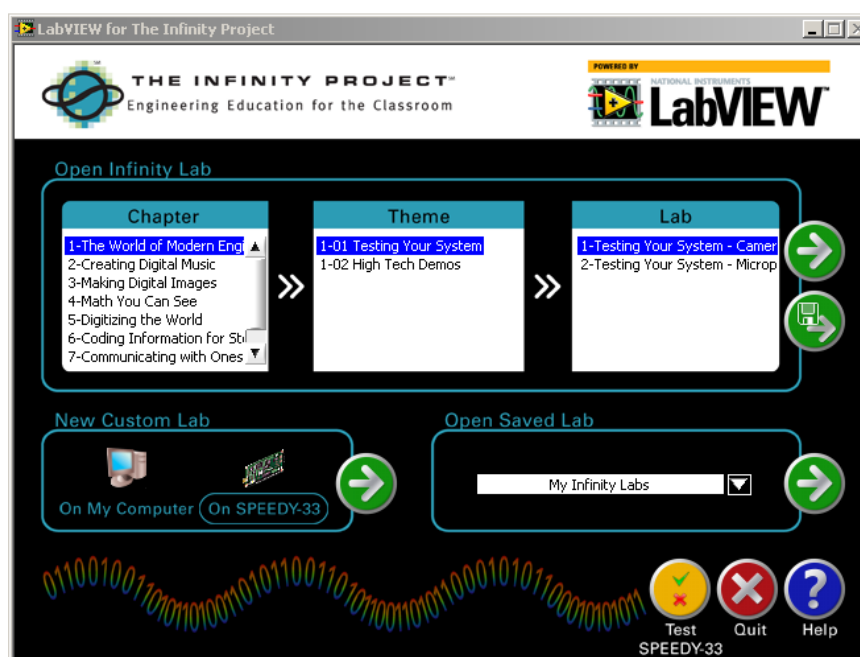
Jedna je označena „Functions“ a nabízí dostupné funkce, jež reprezentují dostupné knihovny. Jsou to např. struktury, generátory funkcí, matematické a logické operace. Jednotlivé bloky můžeme dělit do tří skupin podle charakteru zapojení v diagramu, na zdrojové (hodnota z potenciometru), cílové (vykreslení do grafu) a uzlové, jež jsou bloky, které zpracovávají signál (např. funkce sinus) a mají vstupní a výstupní terminály. Uživatel může vytvořit i vlastní specifickou funkci, kterou poté „vloží do obálky“ a vznikne takzvané SubVI. SubVI je v diagramu prezentováno jako samostatný blok, jež se chová stejně jako předdefinované funkce.

Druhou nabídkou je paleta nástrojů, která nabízí např. brakepoint, sondy, přemístění bloku a wire. Při výběru nástroje se kurzor myši změní na ikonu, která odpovídá aktuálnímu výběru. Při návrhu se tyto nástroje automaticky mění. Jedním z hlavních nástrojů je zde „wire“, jež se používá na propojení jednotlivých bloků. Spojnice se od sebe liší typem čáry a jeho barvou

spočívá v přednostní nabídce funkcí, které jsou určeny pro zpracování obrazu a signálu. Softwarové vybavení je tak určitou podmnožinou LabVIEW dané verze. Dalšími periferiemi, se kterými výukový Infinity projekt počítá, jsou reproduktory, USB kamera, knižní publikace s obecně popsanou problematikou a v neposlední řadě internetové stránky.

Příložená verze SW má v sobě vestavěný DSP modul, který obsahuje funkce, jež zpřístupní hardware kitu. Z toho je zřejmé, že tento modul bude pro práci s NI Speedy – 33 stěžejní součástí a proto mu bude věnována následující podkapitola.

Infinity projekt tak představuje ucelený komplet, který je možné ihned použít. Při spuštění této verze LabVIEW se uživatelům představí nestandardní úvodní okno, viz obr. 12. [11][15][39][13]



obr. 12. LabVIEW Infinity Project

Z úvodního okna jsou zřejmé možnosti spuštění, modifikace nebo založení nového projektu. Nechybí zde ani test komunikace mezi prostředím a HW prostředky kitu.

2.4.1 LabVIEW DSP modul

Tento modul je souborem funkcí určen pro NI speedy-33 a tři kity TI-DSK. Tyto funkce jsou přidány do standardních funkcí LabVIEW a jsou určeny pro vytváření aplikací pro zpracování signálů, které běží na DSP. Použití jazyka G napomáhá k aplikaci metodiky „rapid prototyping“ a rychlého uvedení výsledného výrobku do výroby.

Všechny funkce Labview DSP modulu jsou dostupné z nabídky při vytváření blokového diagramu VI. Lze je rozdělit do několika skupin, kterými jsou: vstupy/výstupy, bloky signálových generátorů, zpracování signálu a moduly pro ovládání motorů.

Bloky vstupů a výstupů umožní použití jednotlivých I/O hardwaru. Můžeme je dělit na analogové a digitální. Analogové vstupy mají tři nastavení - vzorkovací frekvenci, velikost rámce a zesílení. U analogových výstupů lze nastavit pouze vzorkovací frekvenci. Digitální vstupy kitu jsou reprezentovány osmi přepínači a výstupy jsou realizovány osmi LED diodami. Další funkcí je I/O digitální banka, do které je možné ukládat či následně číst digitální signály z DSP.

Další část modulu zahrnuje řadu funkcí pro generování signálu. Jsou zde bloky pro generátory tvaru signálu jako je např. trojúhelník, pila či sinusové průběhy. Je možné také vytvářet signály impulzní, konstantní a „podobné“ šumu.

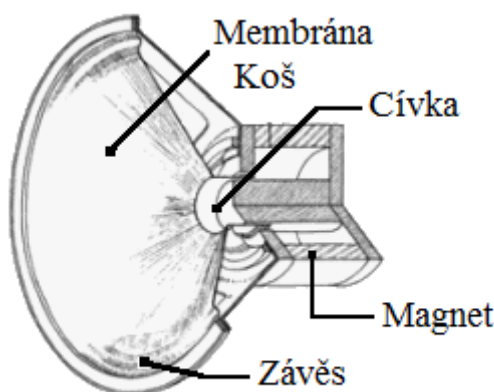
Pro tuto diplomovou práci budou stěžejní bloky, jež se týkají zpracování signálu. Nabídka funkcí je rozdělena do čtyř kategorií – na filtraci, návrhy filtrů pomocí oken, frekvenční a časovou analýzu.

Posledním blokem funkcí, které jsou u DSP modulu k dispozici, slouží pro ovládání motorů. Je zde funkce pro řízení otáček DC motorů (i směru chodu), generování PWM a ovládání servo motorů. [11][15][39][13]

3 Analýza obvodů v reproduktorových soustavách

3.1 Fyzikální vlastnosti reproduktorů

Reproduktory jsou elektroakustické měniče, které přeměňují elektrickou energii na teplo a mechanické kmitání. Reprodukter se skládá z elektrické části, kterou zastupuje kmitací cívka, a mechanické, jež představuje membránu uloženou v koši reproduktoru. Membrána je spojena s košem (pomocí pružného závěsu) a kmitací cívkou obr. 13.



obr. 13. Řez reproduktoru

Cívka je umístěna ve stacionárním magnetickém poli. V případě, že cívkou začne protékat proud, vznikne na vodičích cívky silové působení a dojde k vychýlení membrány. Z popisu principu je zřejmé, že se jedná o značně nelineární systém. Při průchodu střídavého napětí začne docházet ke kmitání cívky, potažmo membrány. Tento pístový pohyb má za následek změny tlaku před membránou. Tímto způsobem vzniká zvuková vlna. V této souvislosti má lidské ucho jednu zajímavou vlastnost. Je amplitudově, frekvenčně nelineární a změnu tlaku vnímá jako zvuk. Zajímavostí je, že na „konstantní“ změny nereaguje (např. změna počasí), ale na rychlé změny ano (zvuk). To je určitým nebezpečím u kvalitních reproduktorových soustav v poslechovém prostředí. V případě, že u takovéto soustavy bude postupně neustále zvyšována úroveň hlasitosti (akustický výkon), adaptivní lidské ucho se této skutečnosti bude přizpůsobovat a může snadno dojít k poškození sluchu. V případě použití reproduktorů s horšími charakteristikami, bude při vyšších hlasitostech patrné zkreslení.

Pro generování zvuku ve slyšitelném spektru (20Hz-20kHz) neexistuje univerzální fyzikální reproduktor. Je to dáno hlavně vlastnostmi přenášených frekvencí. Hlubokotónové reproduktory jsou zpravidla větších rozměrů a použitím dostatečně tuhé membrány (z důvodu její deformace s následky zkreslení). To má za následek vyšší hmotnost pohyblivých částí a vzrůstá tak setrvačnost. Při vyzařování vyšších kmitočtů, kde jsou výchylky membrány rychlejší, proto může docházet u basových reproduktorů k velkému zkreslení. Důležitou roli spjatou s generovanou frekvencí hraje i amplituda. Tyto dva parametry, frekvenci a amplitudu, spojuje frekvenční charakteristika. Jedná se o závislost vyzářené amplitudy na frekvenci. Mnohdy je tento údaj doplněn o tuto charakteristiku měřenou v různých úhlech (mimo osu reproduktoru),

což nám může napovědět i o směrové charakteristice vyzařování reproduktoru. Zpravidla čím má reproduktor větší plochu membrány, tím začíná dříve směřovat (zkreslení SPL). U hifi reproduktorových soustav je kladen důraz na to, aby směrová charakteristika byla co možná největší, a tak poskytla posluchačům jednotný „zážitek z poslechu“.

Dalším parametrem je výkon reproduktoru, od něhož se odvíjí akustický výkon. Akustický výkon P_a je přenesen kruhovou membránou, která kmitá výchylkou x a kmitočtem f . Je možné ho vypočítat z následujícího vztahu:

$$P_a = \frac{\pi^2 r^4 x^2 \omega^4}{2\pi c_0} = 0,856 S^2 x^2 f^4 \quad (1)$$

Akustický tlak p ze vzdálenosti d bude dán:

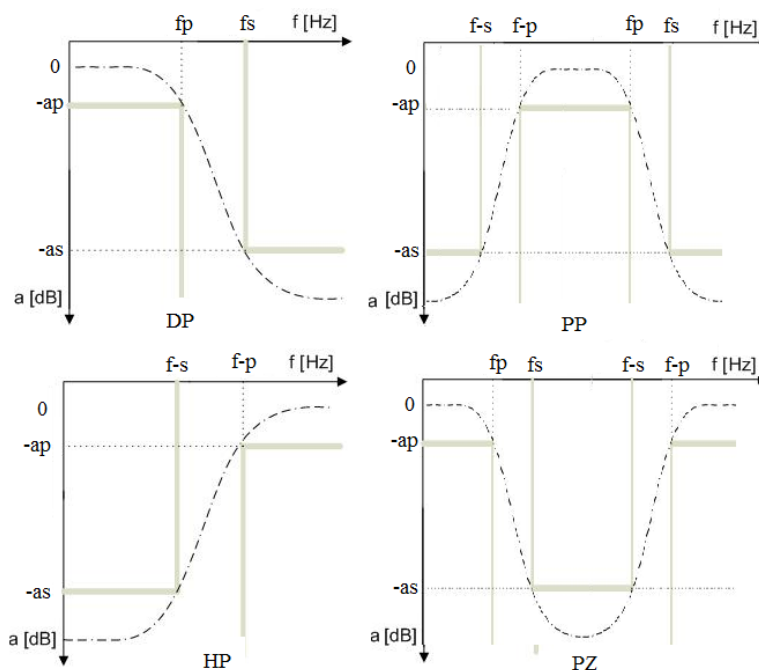
$$P = \frac{2\pi S x f^2}{d c_0} \quad (2)$$

Činitel $x \cdot f^2$ znamená, že akustický tlak je úměrný zrychlení membrány. Konstanta 0,856 je vypočtena pro parametry vzduchu za běžných podmínek. Tento vztah počítá s efektivní výchylkou, jejíž hodnota je pro harmonický signál $1/\sqrt{2}$ – násobkem výchylky maximální. Z výrazů je zřejmé, že pro danou efektivní výchylku x je při nízkých kmitočtech vyzářený výkon úměrný druhé mocnině plochy S a čtvrté mocnině kmitočtu f .

Tento vztah napovídá, že věrnou reprodukci zvuků není možné dosáhnout za pomoci jednoho reproduktoru. Proto musíme použít více „pásen“ rozdělenými frekvenčními výhybkami – filtry. [16][18]

3.2 Typy filtrů

Filtr je obvod, který signály obsažené v určitých frekvenčních pásmech propouští, zatímco jiné potlačuje. Z průběhu amplitudové frekvenční charakteristiky můžeme rozdělit filtry na čtyři základní typy, které jsou vyobrazeny na následujícím obr. 14.



obr. 14. Typy filtrů rozdělené podle frekvenční charakteristiky

Na obrázku je vidět amplitudové-frekvenční charakteristiku, kde na ose x je vyznačena frekvence a ose y pak náleží amplituda. Indexy p a s u frekvence značí toleranční pásmo filtru, kde f_p je ekvivalentní zlomové frekvenci filtru (pokles amplitudy o 3dB). Z obrázku je dále patrné, že základním blokem jsou filtry typu dolní a horní propusti. Vzájemným složením těchto filtrů můžeme vytvořit pásmovou propust nebo pásmovou zadrž.

Filtr je dále charakterizován typem aproximace (např. Buterworthova, Besselova nebo Čebyševova). Ta udává spolu s činitelem jakosti zvlnění amplitudové a frekvenční charakteristiky, jež má ležet v tolerančním pásmu.

Posledním důležitým faktorem je řád filtrů. Ten udává pokles amplitudy na oktávu mezi jeho propustným a nepropustným pásmem. Jeden řád odpovídá poklesu o 6dB/oktávu, další řády jsou násobky této hodnoty (12dB, 18dB...). Oktáva je definovaná jako dvojnásobný kmitočet od zlomové frekvence filtru. Ideální filtry by tak byly reprezentovány nekonečně velkým řádem.

Filtry můžeme dělit dle metody zpracování na analogové a digitální. Analogové filtry užívají spojitých signálů, kdežto digitální pracují s diskrétními signály.

Další častou kategorizací filtrů je dělení dle použitých součástek. V tomto ohledu se jedná o filtry aktivní a pasivní. [7][8][16][18]

3.2.1 Analogové

Analogové filtry (výhybky) jsou reprezentovány jak ve své pasivní formě, která je běžně používaná v Hi-Fi reproduktorových soustavách, tak i v aktivní podobě. Někdy se pasivním výhybkám říká výkonové, protože pracují za zesilovacím prvkem audio sestavy s výkonovým signálem. Takto navržená výhybka více zatěžuje zesilovač případnými ztrátami a navíc má zátěž indukční nebo kapacitní charakter. To může mít za následek zkreslení zesilovače v závislosti na frekvenci. Výkonové výhybky jsou realizovány pomocí pasivních součástek (tj. rezistorů, cívek a kondenzátorů) a zesílení mají nižší než jedna.

Naproti tomu aktivní filtry obsahují zesilovací prvek a jejich zisk tak může být vyšší než jedna. Hlavní výhodou aktivních filtrů je skutečnost, že umožňují realizaci dané přenosové funkce bez použití indukčností. Cívka je totiž prvek, který je obvykle charakterizován velkými rozměry (zvláště pak při filtraci nízkých kmitočtů, kdy je potřeba velkých indukčností), vysokou cenou a při použití feromagnetického materiálu je značně nelineární. Z těchto důvodů se při návrhu filtrů snažíme použití indukčností omezit na co nejmenší míru. Další předností aktivních výhybek je práce s nízkou úrovní signálu a lepší možnost impedančního přizpůsobení řetězce. Tento typ výhybek se užívá převážně v profesionálních reproduktorových soustavách, které přenášejí zpravidla velký výkon. Určitou nevýhodou u aktivních filtrů je nutnost použití stejného počtu výkonových zesilovačů jako je počet zpracovávaných kanálů. [7][8][16][18]

3.2.2 Digitální

Tyto filtry pracují se signály, které jsou vyjádřeny v digitální formě (označení digitální vzniklo z anglického slova digit – číslice) v podobě binárního kódu jedniček a nul. Posloupnost binárního kódu pak určuje tvar signálu. To umožňuje zpracování signálu pomocí počítače, DSP či mikrokontroléru.

I přesto, že původní analogový signál je třeba převést do digitální formy, při níž dochází k určitým ztrátám kvality původního signálu, je digitální zpracování signálu hojně užíváno v moderních audio-vizuálních systémech.

Při návrhu digitálního filtru můžeme užít různých struktur návrhů při zachování stejných vlastností digitálního filtru v kmitočtové oblasti.

Základní rozdělení struktur je možné rozčlenit z hlediska zpětné vazby na:

- filtry rekurzivní (RDF), které zpětnou vazbu ve své struktuře mají
- a filtry nerekurzivní (NRDF), které neobsahují ve svém blokovém schématu zpětnou vazbu.

Další klasifikace pak vychází z impulzní odezvy digitálního filtru. Tu můžeme rozdělit na filtry:

- IIR filtry – s nekonečnou impulzní odezvou
- FIR filtry – s konečnou impulzní odezvou

Následnou metodou dělení struktur by mohlo být podle uspořádání funkčních „bloků“ na:

- přímé
- kaskádní
- paralelní

U nerekurzivních filtrů je hodnota výstupního vzorku závislá pouze na vstupních hodnotách. To zajistí i konečnou odezvu impulzní charakteristiky, a proto jsou nerekurzivní filtry vždy typu FIR. U rekurzivních filtrů je výstupní hodnota závislá nejen na jeho vstupních hodnotách, ale také na hodnotách výstupních. Struktura obsahuje alespoň jednu zpětnou vazbu. Rekurzivní strukturu užívají vždy filtry typu IIR.

Užití přímých struktur je nevýhodné vzhledem k velké citlivosti vlastností filtrů na přesnost vyjádření koeficientů. I malá změna jednoho koeficientu způsobí přesunutí nul a pólů, tím i změnu frekvenční charakteristiky daného filtru. Tento vliv je možné částečně eliminovat rozdělením přenosové funkce $H(z)$ do většího počtu dílčích přenosových funkcí $H_i(z)$, kde každá má nižší počet nul a pólů. Výsledná přenosová funkce $H(z)$ je pak realizována pomocí kaskádního nebo paralelního spojení jednotlivých dílčích přenosových funkcí $H_i(z)$. [3][8][9][10]

IIR

IIR jsou digitální filtry, které mají svou obdobu u filtrů analogových. Lze s nimi realizovat Chebyshevovu, Butterworthovu či Caurovú aproximaci filtru. IIR filtry jsou rekurzivní a jsou definovány skupinou rekurzivních činitelů. To znamená, že využívají předem vypočítaných hodnot na výstupu a obsahují alespoň jednu zpětnovazební smyčku. Jednou ze základních vlastností těchto filtrů je jejich impulzní charakteristika, která je složená ze sinusoid, kde exponenciálně klesá amplituda. Tento fakt zapříčiní nekonečně dlouhou impulzní odezvu. Proto se tyto filtry nazývají „Infinite Impulse Response“ (IIR). IIR filtry mají oproti FIR filtrům velký poměr mezi propustným a nepropustným pásmem. Z tohoto je zřejmé, že pro stejnou aproximaci frekvenční charakteristiky mají výrazně nižší řád než filtry FIR. Ve výsledku jsou tyto filtry jednodušší, a tím i méně náročné na požadavky hardwaru. Jednoduchost je však vykoupena nelineární fází a při špatném návrhu mohou být nestabilní.

Návrh IIR filtrů můžeme popsat následující rovnicí:

$$y[n] = \sum_{k=0}^n b_k x[n-k] - \sum_{k=0}^n a_k x[n-k] \quad (3)$$

[3][8][9][10]

Butterworthova aproximace

Pro návrh IIR filtrů bude v práci užita Butterworthova aproximace. Butterworthův filtr se vyznačuje specifickým polynomem, jehož kořeny leží v komplexní rovině. Koeficienty normovaného Butterworthova polynomu jsou voleny tak, aby při zlomové frekvenci filtru byla ω_c právě 1 [rad/sec]. Všechny jeho kořeny leží v levé části poloroviny, a tím je zajištěna stabilita systému. Pro lichý řád je jeden koeficient vždy reálný a roven -1, další kořeny jsou rovnoměrně rozloženy na jednotkové kružnici. Tuto hlavní charakteristiku můžeme popsat následujícím vztahem:

$$|H(j\omega)| = \frac{b_0}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}} \quad (4)$$

Z tohoto vztahu je zřejmé, že jakýkoli řád filtru bude mít zesílení $b_0/\sqrt{2}=0,707$ pro frekvence rovnající se frekvenci zlomové. Tato hodnota odpovídá poklesu charakteristiky o 3dB. Frekvenční odezvu můžeme získat z kvadrátu předchozího vztahu:

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{j\omega}{\omega_c}\right)^{2n}} ; j = \sqrt{-1}, \quad (5)$$

kde ω je frekvence signálu a ω_c značí zlomovou frekvenci. V případě, že $\omega=0$ bude funkce nabývat hodnoty 1 a vstupní signál bude plně propuštěn. V případě, že se $\omega=\infty$, bude signál zcela zadržen a funkce bude nabývat hodnoty 0. Vyjádříme-li přenosovou funkci v Laplaceově transformaci s proměnou s , která je komplexně sdružená, přejde rovnice na tvar:

$$H(s)H(-s) = \frac{1}{1 + \left(\frac{s}{\omega_c}\right)^{2n}} ; s = j\omega \quad (6)$$

Z této rovnice můžeme vyjádřit polohu jednotlivých pólů:

$$1 + \left(\frac{s}{j\omega_c}\right)^{2n} = 0 \quad (7)$$

$$\left(\frac{s}{j\omega_c}\right)^{2n} = -1 \quad (8)$$

$$\frac{s}{j\omega_c} = (-1)^{\frac{1}{2n}} \quad (9)$$

$$s = (-1)^{\frac{1}{2n}}(j\omega_c) \quad (10)$$

Z tohoto vztahu můžeme dále vyjádřit obecný tvar pro výpočet kořenů:

$$S_k = \omega_c * e^{j\pi(2k+1)/2n}, \quad (11)$$

kde $n=2,4,6\dots$ a $k=0,1,2\dots,2n-1$. Z předchozích poznatků je možné zapsat vztah frekvenční odezvy:

$$H(s) = \frac{1}{\frac{s}{j\omega_c} - e^{j\frac{3\pi}{4}}} * \frac{1}{\frac{s}{j\omega_c} - e^{j\frac{5\pi}{4}}} = \frac{\omega_c}{s - e^{j\frac{3\pi}{4}\omega_c}} * \frac{\omega_c}{s - e^{j\frac{5\pi}{4}\omega_c}} \quad (12)$$

$$H(s) = \frac{\omega_c^2}{s^2 - 2\cos\left(\frac{3\pi}{4}\right)s + \omega_c^2} = \frac{\omega_c^2}{s^2 + 2\cos\left(\frac{\pi}{4}\right)s + \omega_c^2} \quad (13)$$

[34][34][33][17][12]

Po bližším prozkoumání můžeme odvodit normalizované vztahy Buttworthovy aproximace, kdy zlomový kmitočet bude nabývat hodnot právě 1[rad/sec]. V tomto případě se bude frekvence vstupního signálu rovnat zlomové frekvenci filtru. Pro aplikování této aproximace pro bikvadratické sekce je možné uvést tyto normalizované tvary [17][8][9]:

- Pro první řád:

$$H(s) = \frac{1}{s + 1} \quad (14)$$

- Pro každý sudý řád N :

$$H(s) = \prod_{n=1}^{N/2} \left(\frac{1}{s^2 + \alpha_n s + 1} \right), \quad (15)$$

kde

$$\alpha = 2\cos\left(\frac{\pi}{2N}(2n-1)\right) \quad (16)$$

- Pro každý lichý řád N větší než jedna:

$$H(s) = \frac{1}{s+1} \prod_{n=1}^{\frac{N-1}{2}} \left(\frac{1}{s^2 + \alpha_n s + 1} \right), \quad (17)$$

kde

$$\alpha = 2\cos\left(\frac{n\pi}{N}\right) \quad (18)$$

FIR

FIR jsou digitální filtry, které nemají obdobu u filtrů analogových. Jsou složitější na návrh a náročnější na jejich výpočet. Mají výrazně nižší útlum mezi propustným a nepropustným pásmem. Proto je při návrhu nutné použít mnohonásobně větší řád než u IIR filtru. Tyto filtry jsou nerekurzivní (nemají tedy zpětnou vazbu) a oproti zmíněným IIR jsou stabilní s lineární fází. Lze tedy říci, že jsou přesnější. Za pomoci FIR je možné realizovat filtry s libovolnými vlastnostmi. Pracují na principu konvoluce vstupního signálu s výstupní charakteristikou. Impulzní charakteristika se zde nazývá jádro filtru (neboli kernel). Mezi filtry FIR se řadí například Moving average, vícepropustný Moving average, Gaussův a Blackmanův filtr atd. FIR filtry můžeme popsat následující rovnicí: [12][10][8][7]

$$y[n] = \sum_{k=0}^n b_k x[n-k] \quad (19)$$

3.2.3 Srovnání číslicového a analogového zpracování signálů

Srovnání základních faktů mezi číslicovým a analogovým zpracováním bude popsáno z hlediska číslicového zpracování. Tyto metody mají oproti analogovému zpracování jisté omezení, které vnáší do obvodu A/D převodník.

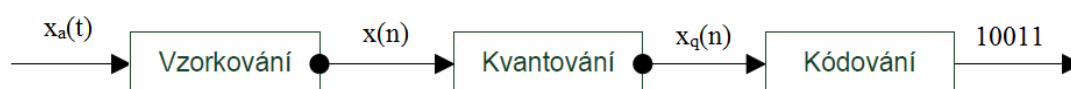
A/D převodník číslicového obvodu udává rozsah zpracovávaného kmitočtu signálu. Zpracování signálu číslicovými obvody je vhodné pro nižší frekvence (stovky MHz). Tento fakt je dán hlavně cenovou dostupností. Existují samozřejmě i převodníky rychlejší, avšak s malou rozlišovací schopností. Například A/D převodník „AT84AS008“ firmy Atmel zvládá vzorkovací frekvenci 2,2 GHz při hloubce 10ti bitů. Jeho cena se však pohybuje kolem 2-3 tisíc USD. Další nevýhodou je nezvratná degradace analogového signálu jeho navzorkováním. Ztratí se tak jeho původní spojitost.

V případě, že např. měřicí systém nepracuje s vysokými frekvencemi a dostačují tak běžné A/D převodníky, můžeme použít číslicové zpracování se všemi jeho výhodami. Nesporným kladem digitálního zpracování je pružnost případných změn. Zatímco u analogové dolní propusti by se při změně musely měnit hardwarové prvky (odpory, kondenzátory, cívky), tak u dolní propusti realizované pomocí DSP by stačila změnit určitá konstanta v programu. Jednoduchá změna konstant v programu tak umožňuje adaptivní filtrování. Je samozřejmostí, že použití číslicových obvodů s sebou nese i určitou přenositelnost algoritmů mezi různým HW změnou řídicího programu. Další výhodou je jeho časová stálost, která je dána zpravidla konstantami a programem. Pokud takový číslicový systém funguje, nedochází ke změně jeho charakteristik vlivem např. teploty či okolního prostředí a jsou také dokonale reprodukovatelné. Při mnoha procesech zpracování hraje roli zaznamenávání stavů či měřených průběhů do paměti. Realizace analogové paměti je příliš složitá, s čímž jsou samozřejmě spojené i vysoké náklady. Digitální záznam je oproti analogovému zpravidla stálejší, spolehlivější, celkově jednodušší. V mnoha případech také odpadají problémy spojené se vzájemným rušením jednotlivých systémů např. impedanční přizpůsobení či vliv elektro-magnetického pole. Dalšími vlastnostmi je samozřejmá slučitelnost s informačními systémy, která v dnešní době hraje značnou roli.

Z předchozího textu vyplývá, že A/D a D/A převodník je důležitým prvkem při zpracovávání analogových signálů v digitální podobě a proto bude v následující kapitole rozebrána jeho problematika. [4][1][5][7][3]

3.3 Analogo – digitální a digitálně - analogová konverze

Mnoho signálů, se kterými je potřeba pracovat, jako je řeč, seismické signály, radarové signály, sonarové signály nebo komunikační signály jako je audio či video, jsou analogové. Pro práci s analogovými signály v digitální formě musíme nejprve mít jejich posloupnost čísel s konečnou přesností. To obstarává tzv. A/D převodník. A/D převod je možno rozdělit do tří základních bloků, viz obr. 15.[3][1][5][7]



obr. 15. Analogo-digitální konverze

Kvalitu A/D a D/A převodníku můžeme definovat dvěma parametry. Tím prvním by byla vzorkovací frekvence a tím druhým počet bitů použitého převodníku. Vzorkovací frekvence nám udává maximální možný přenesený kmitočet vstupního signálu, zatímco bitová hloubka nám udává popis amplitudy signálu. Bitovou hloubku je možné představit např. pomocí hloubky barev, viz obr. 16.

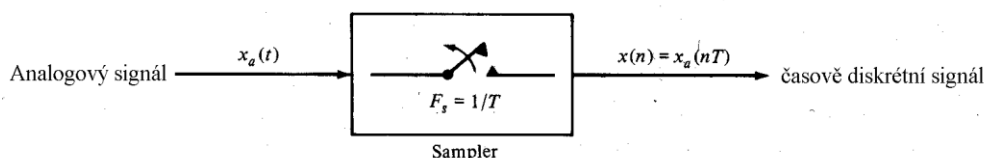


obr. 16. Bitová hloubka barev, vlevo 256barev, v pravo 16mil. barev

Po zpracování signálu číslicovým obvodem je většinou potřebná zpětná rekonstrukce na analogový signál. To má za úkol D/A převodník, který z posloupnosti čísel vytvoří „diskrétní“ analogový signál. Průběh tohoto signálu bude spojitý, avšak amplituda bude schodkovitého „diskrétního“ charakteru. Proto je nutné za D/A převodník vřadit rekonstrukční filtr. Úkolem tohoto filtru je odstranit schodkové průběhy vyšších kmitočtů výstupního signálu. Tím se docílí spojitosti amplitudy výstupního signálu. V praxi je tedy použit dolnoproustný filtr, který je navržen s ohledem na Shanon-Kotelníkův teorém. Zlomový kmitočet filtru tedy musí být menší než polovina vzorkovací frekvence. [4][1][5][7]

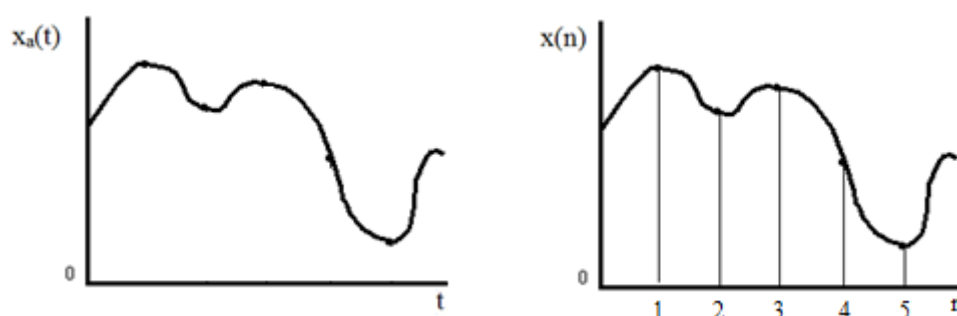
3.3.1 Vzorkování

Vzorkování je proces, kdy z časově spojitého signálu jsou získány vzorky „samples“, které tvoří časově diskrétní signál. Jestliže na vstupu vzorkovacího obvodu bude signál $x_a(t)$, pak na jeho výstupu bude signál $x_a(nT) \equiv x(n)$, kde T je vzorkovací perioda. Jedná se tedy o signál, který je v čase již nespojitý, ale v amplitudě může dosáhnout kteroukoli hodnotu z původního spojitého signálu. [2][3][7][8]



obr. 17. Vzorkovací blok

Vzorkovací frekvence se pak vypočte $f_v = 1/T$. Pokud je kmitočtový rozsah vstupního signálu $x_a(t)$ příliš velký, jsou i jeho časové změny příliš rychlé a tak může docházet k převodu, který bude zatížen chybami. [3][2][7][8]



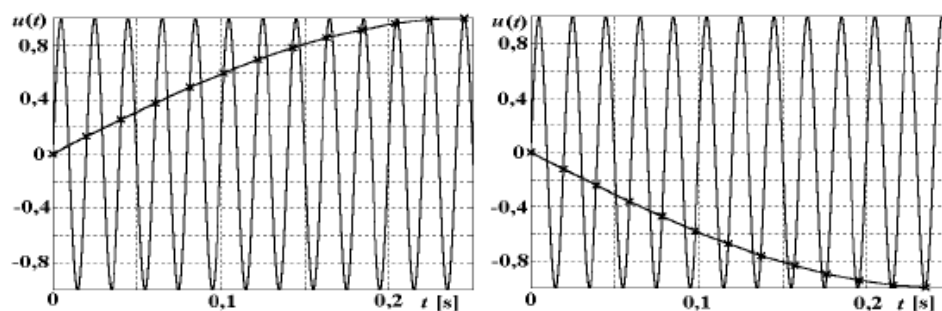
obr. 18. Vzorkování signálu

Shanon - Kotelníkův teorém

Někdy je označován také jako Nyquistův teorém. Říká, že rekonstrukce původního spojitého, frekvenčně omezeného signálu z jeho vzorků je možná tehdy, pokud byla vzorkovací frekvence alespoň dvakrát vyšší než maximální frekvence rekonstruovaného signálu. V praxi se vzorkovací frekvence užívá o něco vyšší než dvojnásobná. Např. u nosičů CD, kde se předpokládají slyšitelné frekvence až 20kHz, se používá vzorkovací frekvence 44,1kHz. Při nedodržení Shanon – Kotelníkova teorému dochází k tzv. aliasingu. [10][12][2][31]

Antialiasing

Aliasing se projevuje u každého zařízení nebo procesu, kde se spojitá funkce reprezentuje pomocí diskrétních vzorků. Lze jej považovat za určitý druh zobrazení interferenčního vlivu, kde existuje určitý druh poměru mezi vzorkovací frekvencí a opakováním struktury dat. Aliasing vzniká při nedodržení Shanon-Kotelníkova teorému, kdy se vysoké frekvence „překládají“ do nižších, viz obr. 19.

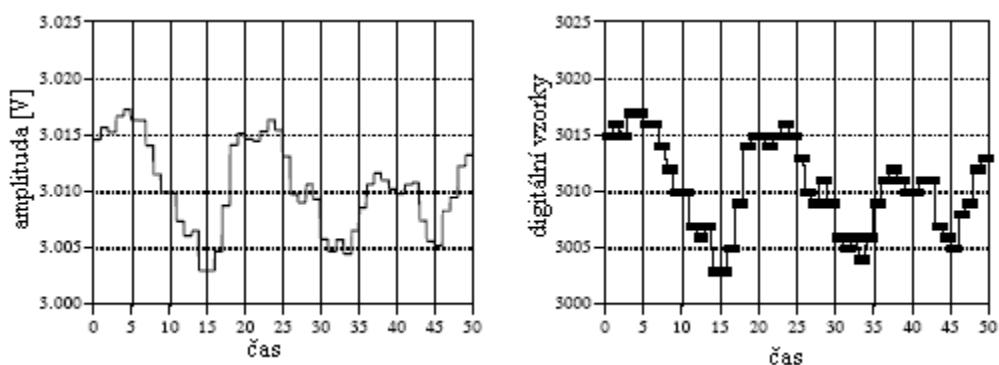


obr. 19. Vznik aliasingu – signál o frekvenci 50Hz je snímán 49Hz (vlevo) a 51Hz (vpravo)

Aliasingu je možné jednoduše zabránit vložením předřadného dolnoproustného filtru před A/D převodník. Tento filtr se nazývá „antialiasing“ a má za úkol propustit pouze frekvence, které se bezpečně správně navzorkují. [10][12][2][31]

3.3.2 Kvantování

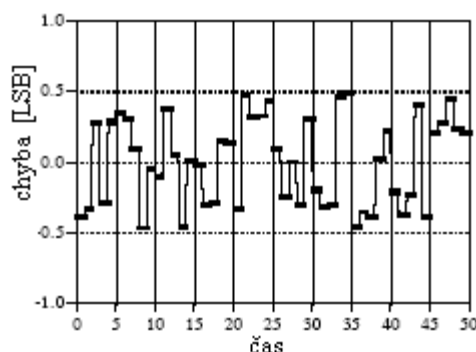
Jedná se o proces, kdy časově diskretní signál převedeme na diskretní digitální signál. Jde tedy o výběr určitých omezených počtů amplitudových stavů, které mohou být vyjádřeny určitým počtem bitů k dané amplitudě. Tuto důležitou úlohu provádí operace - kvantování vzorků. Z toho vyplývá, že pro přesnější amplitudovou rekonstrukci je třeba „jemnější“ A/D převodník s více možnými hladinami. Počet hladin určují bity převodníku, jejichž hodnotu můžeme vypočítat 2^n , kde n je počet bitů A/D převodníku. Na obr. 20 je příklad kvantování 12 bitovým A/D převodníkem, který má 4096 úrovní, kde každá úroveň odpovídá 1mV. [3][2][31][7]



obr. 20. Vzorkovaný a kvantovaný signál

Rozdíl mezi hodnotou časově diskretního signálu $x(n)$ a digitálního signálu $x_q(n)$ se říká kvantizační chyba. Tato chyba vzniká vlivem prahových kvantizačních úrovní, kdy je tedy původní signál zaokrouhlen na určitou hodnotu a vzniká tak rozdíl mezi navzorkovaným a

nakvantovaným signálem (obr. 21). Pokud tuto chybu zaneseme do grafu, získáme tzv. „kvantizační šum“, který je nápadně podobný náhodnému šumu. [10][12][2][31]



obr. 21. Kvantizační chyba

Jinými slovy, analogový vstup je ekvivalentní digitálnímu výstupu s kvantizační chybou. Chyba je zobrazena v tzv. LSB, což je „nejméně významný bit“. Kvantizační chyba je u všech lineárních převodníků stejná a spadá do intervalu $+1/2$ až $-1/2$ kvantizační úrovně. Protože odstup signálu od šumu je zvykem zapisovat v podobě poměrného čísla v decibelech, můžeme zavést vztah: [2][4]

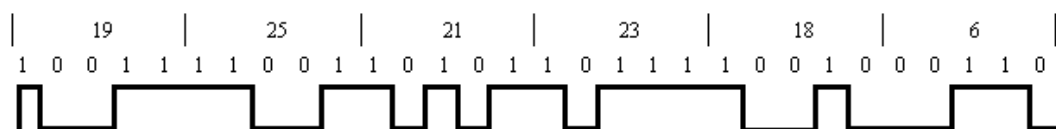
$$SNR_{A/D} = 20 * \log 2^N \approx 6,02 * N[dB], \quad (20)$$

kde N je počet bitů A/D převodníku. Pro příklad: hudební CD je vzorkováno 44,1KHz s bitovou hloubkou 16bit, jeho odstup signálu od šumu je $16 * 6,02 = 96,32$ dB. [10][12][2][31]

3.3.3 Kódování

Vzorkováním a kvantováním se dosáhne nahrazení spojitého signálu omezeným počtem kvantovaných vzorků. Takto upravený digitální signál je odolný rušení a lze ho popsat dvěma osami, kdy na jedné by byla vynesena vzorkovací frekvence a na druhé amplituda. Je zřejmé, že operace kvantování a kódování jsou i po technické stránce úzce svázané.

Kódování je možno provádět více způsoby, ale nejčastěji se používá kódování po bitech, zvané také váhování. Jeho podstatou je postupné amplitudové porovnávání napětí vzorků signálu s napětím z napěťového děliče komparátoru. Napěťové děliče obsahují N odporových článků, které v obvodovém zapojení představují zdroje stabilizovaného napětí. Výstupní napětí z jednotlivých článků mají svou určitou velikost - váhu, která odpovídá vahám jednotlivých bytů kódové skupiny. Například dvojkové číslo: 1 0 0 1 1 s vahami míst 16 8 4 2 1 má v desítkové soustavě velikost $16 + 2 + 1 = 19$ viz obr. 22. [10][12][2][31][2][3]



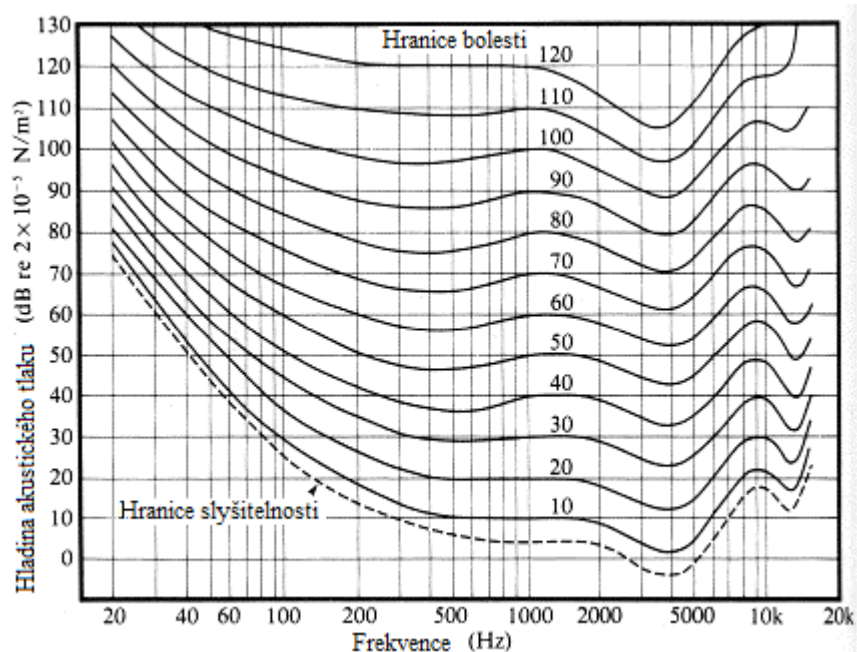
obr. 22. Příklad 5bit kódování

3.4 Aplikace filtrů pro DSP procesor

Z předchozí kapitoly, která se věnovala hardwaru a softwaru kitu NI Speedy – 33, vyplývají možnosti použití. Pomocí NI Speedy – 33 je možné zpracovávat pouze dva vstupní signály, přičemž jeho výstupy jsou limitovány také dvěma kanály. Z toho vyplývá, že je možné pracovat pouze se stereo signály. V případě použití kitu jako dvoupásmové výhybky je možné zpracovávat pouze jeden kanál stereofonního signálu. Toto omezení je možné částečně potlačit použitím operačních zesilovačů. V tomto případě bychom kit použili pro filtraci basových pásem stereofonního signálu. Výstupy jednotlivých pásem by v tomto případě byly přivedeny spolu s původním signálem na vstupy diferenčních zesilovačů. Touto metodou by bylo možné realizovat stereofonní dvoupásmovou výhybku za použití dvou výstupů kitu, avšak za předpokladu, že by původní signál byl vhodně fázově posunut.

Dalším filtrem signálu je parametrický equalizér. Toto označení bylo původně určeno pro zařízení, které se používalo k vyrovnání barvy zvuku v nedokonalých (reprodukčních nebo nahrávacích) systémech. Ekvalizér (zkratkou EQ) představuje tónovou clonu, která je vřazena do audio řetězce. Funkce ekvalizéru by se dala popsat jako zesílení v určitém frekvenčním rozsahu. Běžné ekvalizéry, které můžeme najít v Hi-Fi zesilovačích, umožňují změnit zdvih nebo potlačení signálu určité frekvence, na které jsou naladěny. Méně často je možné najít ekvalizéry, kde je k zesílení možné měnit i zvlnění Q neboli šířku pásma. Hlavním rozdílem parametrického ekvalizéru oproti obyčejnému je ten, že můžeme měnit jeho frekvenční rozsah. V praxi se běžně jedná o přeladitelnou pásmovou propust doplněnou o zesilovací prvek, jež může nabývat jak kladných, tak i záporných hodnot. Dalším parametrem může být typ kmitočtové charakteristiky např. BELL, SHELVING, NOTCH.[16]

Posledním filtrem, který zde bude popsán je ABL (Adaptive Bass Linearization) funkce. Tato funkce má za úkol linearizovat basové spektrum reproduktorové soustavy. Je založen na nelinearitě vnímání lidského ucha, viz obr. 23. [16][17][18]



obr. 23. Fletcher – Monsonovy křivky stejné hlasitosti

Z obrázku je zřejmé, že ucho je značně nelineární při nízkém akustickém tlaku v oblasti basových kmitočtů. Při zvyšování akustického tlaku se pak částečně linearizuje. Další nelinearita lidského sluchu náleží kmitočtu okolo 3500Hz, kde je lidské ucho nejcitlivější. [18]

Právě linearizací v oblasti basových kmitočtů se funkce ABL zabývá. Jde tedy o to, že při nízkém akustickém tlaku budou basové kmitočty zesilovány oproti středním a nižším kmitočtům. S rostoucím vybuzením reproduktorové soustavy bude zesílení funkce ABL klesat. Z popisu je zřejmé, že funkce ABL je určitou obdobou fyziologické regulace, jež je známa z analogových obvodů užívaných ve zvukové technice. [16][18]

4 Návrh aplikace s GUI rozhraním

V této kapitole budou popsány dvě možnosti aplikací digitálních filtrů pro kit NI Speedy – 33. První možností je filtrace vstupního signálu v časové oblasti a druhou je pak filtrace v oblasti frekvenční. V následujících podkapitolách jsou popsány jednotlivé algoritmy, které jsou v této práci použity a za nimi následují jejich GUI rozhraní sloužící k monitorování a konfiguraci jednotlivých filtrů.

GUI bylo navrženo prostřednictvím vývojového prostředí LabVIEW. Pro vyšší přehlednost byla zavedena dvě pravidla, kterých se drží všechny naprogramované úlohy. První pravidlo se týká vykreslování charakteristik, kde vstupní charakteristiky jsou vykreslovány bílou barvou a výstupní žlutou. Druhé pravidlo se týká ovládacích prvků, kde simulované procesy ke svému ovládání používají výhradně vertikálních tahových potenciometrů, kdežto otočné potenciometry jsou přiděleny pro změnu hodnot, které se uplatňují při samotném výpočtu procesů reálného času.

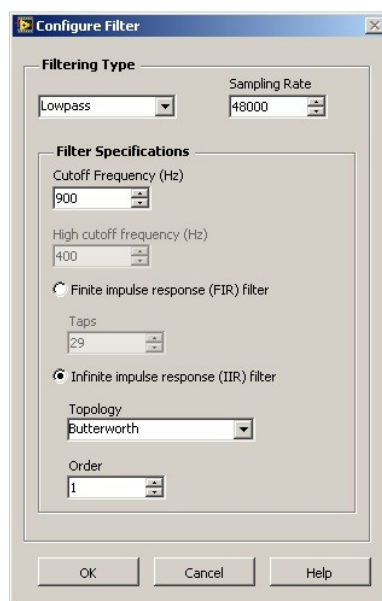
4.1 Filtrace v časové oblasti

Pro filtraci v časové oblasti je možné použít v LabVIEW Infinity Project tři funkce, z čehož jedna ke svému návrhu užívá nástroj NI Digital Filter Design Toolkit, který však není součástí instalace a je licencován. Proto tato metoda nebyla v práci použita. Funkce, které realizace užívá, je možné najít pod názvem bloku „Filter“ a „Biquad“. V případě „Filter“ se jedná o expresní funkci, kterou je možné před kompilací programu nakonfigurovat. V běžícím programu je však konfigurační okno nedostupné a nelze měnit jednotlivé parametry filtru. Posledním blokem, který je možné pro filtraci v časové oblasti použít je tzv. „Biquad“, jež reprezentuje přenos IIR filtru druhého řádu. Výhodou této funkce oproti „Filter“ je možnost změny jednotlivých koeficientů za chodu programu.

Obě zde zmíněné funkce jsou užity při realizaci filtrů a jsou podrobněji popsány v podkapitole 4.1.1 a 4.1.2. [13][2][3]

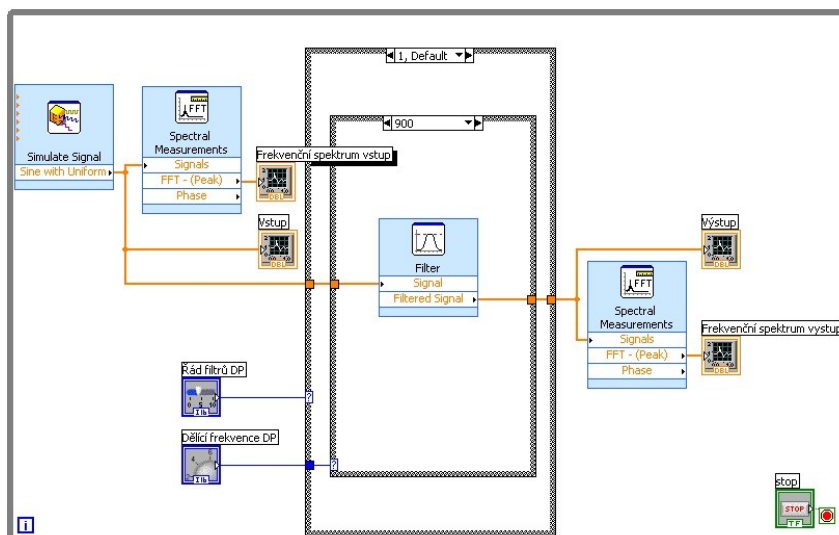
4.1.1 Filtrace užívající expresní funkci

Tato metoda využívá předdefinovaných filtrů, které jsou uloženy v case strukturách, kde jednou case strukturou je volen řád filtru a druhou strukturou je vybrán kmitočet. Algoritmus je pro dolní i horní propust stejný. Jediným odlišením je nastavení jednotlivých filtrů, které je zřejmé z obr. 24.



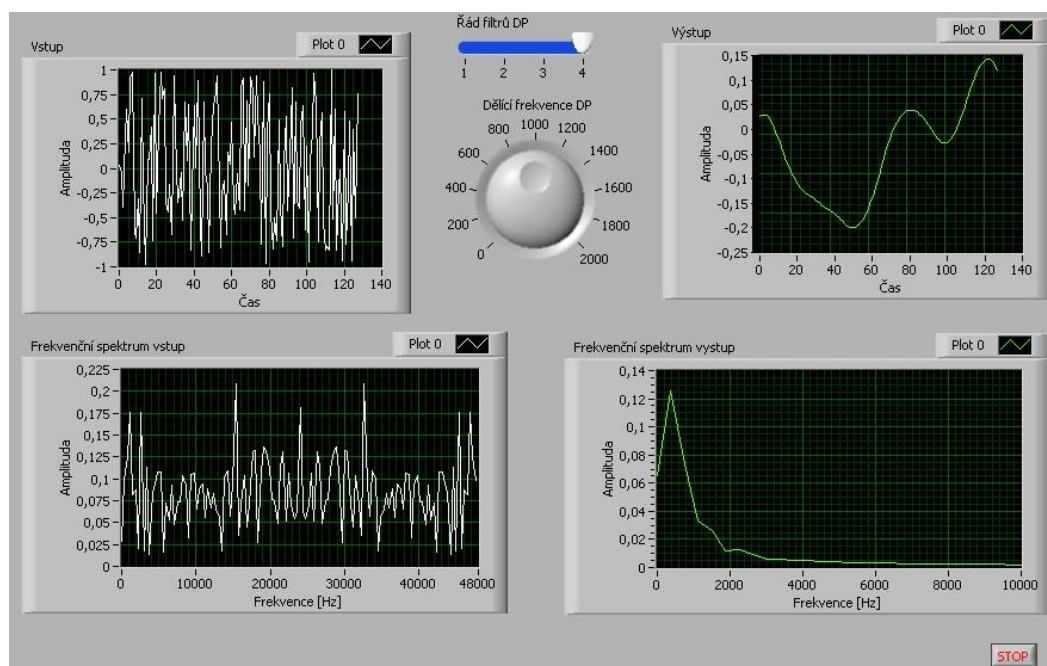
obr. 24. Nastavení expresní funkce Filter

Na obr. 25 je vyobrazen algoritmus dolní propusti 1. – 4. řádu s nastavitelnou frekvencí v rozmezí 100 - 2000Hz s krokem 100Hz. Řád filtru je volen case strukturou, ve které se nachází další vnořená case struktura určující požadovaný kmitočet. Pro generování vstupního signálu je použita funkce „Simulate Signal“, která následně vstupuje do bloku filtru. Pro ověření funkčnosti filtru je na vstupu a výstupu filtru přidán blok spektrální analýzy a veškeré charakteristiky jsou zobrazovány v grafech na hlavním panelu. Všechny použité prvky jsou uzavřeny ve struktuře while, která se vykonává do doby, než je stisknuto tlačítko STOP na hlavním panelu.



obr. 25. Algoritmus DP realizován expres funkcí

Výběr jednotlivých kmitočtů a řádů filtru se provádí změnou hodnoty potenciometru a posuvného přepínače na hlavním panelu aplikace, viz obr. 26. Hodnota posuvného přepínače udává zvolený řád a je přivedena na první case strukturu, která přepíná řády filtru. Potenciometrem je nastavována hodnota dělicí frekvence, která je přivedena na vnitřní case strukturu, jež přepíná jednotlivé přednastavené filtry.



obr. 26. Hlavní panel DP

Na obr. 26 je vyobrazen hlavní panel DP využívající přednastavených filtrů. Ze zobrazených grafů na hlavním panelu je zřejmá funkčnost DP filtru.

Algoritmus horní propusti je totožný s propustí dolní. Jediný rozdíl spočívá v nastavení bloku filtru. Hlavní panel v případě HP vypadá stejně, avšak kmitočtový rozsah je zvolen v rozmezí 2-4kHz. Touto metodou byla naprogramována i pásmová propust, ta však nefungovala kvůli nedostatku paměti HW a proto zde nebude uvedena.

4.1.2 IIR filtr 2. Řádu

Je nazýván také bikvadratický, jelikož je jeho přenos dán dvěma kvadratickými polynomy. První je reprezentován čitatelem a druhý jmenovatelem. Jeho přenosová funkce má následující tvar:

$$H_{(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \quad (21)$$

Blok „biquad.vi“, který je přístupný v LabVIEW DSP modulu, používá následující přenosovou funkci:

$$H_{(z)} = g * \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \quad (22)$$

kde g znamená zesílení filtru, které je reprezentováno separátním blokem (násobením). Koeficienty, ale také zesílení g lze získat z původní rovnice bikvadratického filtru následující úpravou:

$$H_{(z)} = \frac{1}{a_0} * \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + \frac{a_1}{a_0} z^{-1} + \frac{a_2}{a_0} z^{-2}} \quad (23)$$

V předchozí kapitole byly popsány vlastnosti IIR filtrů, ze kterých je zřejmé, že je možné pro jejich návrh využít analogové filtry jako jejich předlohy. Právě její pomocí můžeme jednoduše vyjádřit jednotlivé koeficienty bikvadratického přenosu. Výpočet koeficientů bude proveden pro Butterworthův DP filtr prvního a druhého řádu. V tomto případě bude mít průběh frekvenční charakteristiky klesající charakter, kde číselník přenosu bude konstantní a absolutní hodnota jmenovatele reprezentována polynomem se bude s rostoucí frekvencí zvyšovat.

Pro zjištění koeficientů přenosu filtru v rovině $H(z)$ je možné použít bilineární transformaci, která převádí spojitý systém na diskrétní. [2][3][34][33][32][31][17][10][9][7]

Odvození bilineární transformace je možné provést několika ekvivalentními způsoby. Vzorkování systému můžeme provést tak, že systém popsaný stavovými rovnicemi, ve kterém jsou vzorky reprezentovány v čase t , nahradíme diskrétními časovými okamžiky $t=nT$, kde T je vzorkovací perioda. Vzorkování analogového signálu a Laplaceova transformace vzorkovací funkce vede na ekvivalenci. [2][3][34][33][32][31]

:

$$z^{-1} \sim e^{-sT} \quad (24)$$

Komplexní kmitočet s je možné získat úpravou vztahu (24) na:

$$s = \frac{1}{T} \ln z \quad (25)$$

Pro vyjádření přenosových vlastností rozvineme pravou stranu rovnice v řadu, přičemž v úvahu se bere pouze první člen rozvoje.

$$\ln z \approx 2 \left[\frac{z-1}{z+1} + \frac{1}{3} \left(\frac{z-1}{z+1} \right)^3 + \frac{1}{5} \left(\frac{z-1}{z+1} \right)^5 + \dots \right] \quad (26)$$

Při dosazení vždy prvních výsledků rovnice (26) do (25) získáme tři transformace z do s , které jsou označovány jako BD (backward difference), FD (forward difference) a bilineární transformace. Pro komplexní kmitočty bude následně platit rovnice:

$$j\Omega = \frac{2}{T} \frac{1 - e^{-j\omega T}}{1 + e^{-j\omega T}} = j \frac{2}{T} \frac{e^{j\omega T/2} - e^{-j\omega T/2}}{e^{j\omega T/2} + e^{-j\omega T/2}} = j \frac{2 \sin(\omega T/2)}{T \cos(\omega T/2)} = j \frac{2}{T} \tan \frac{\omega T}{2} \quad (27)$$

Z rovnice vyplývá, že pro všechny ω je absolutní hodnota $|z=e^{j\omega T}|=1$, což je rovnice kružnice se středem o poloměru 1. Z rovnice (27) je možné vyvodit, že kmitočtové osy jsou vzájemně zkreslené a platí:

$$\Omega = j \frac{2}{T} \tan \frac{\omega T}{2}, \quad (28)$$

kde výraz $2/T$ reprezentuje konstantu, která bude v následujících výpočtech nahrazena konstantou c . Tato záměna nemá pro výpočet bilineární transformace žádný vliv. Je však nutné držet tuto konvenci pro celý výpočet i v případě zpětné bilineární transformace. Konstanta c je vypočtena z upravené rovnice (28):

$$c = \cot \left(\pi \frac{f_{cut}}{f_s} \right) = \cot \left(\frac{\omega_{cut} T_s}{2} \right) \quad (29)$$

Posledním krokem pro výpočet koeficientů je mapování roviny s pro horní, dolní propust a pásmovou zádrž. Mapování je pro dolní propust možné popsat následujícím vztahem:

$$s = \frac{S_{\omega_c}}{\omega_c}, \quad (30)$$

pro horní propust pak platí:

$$s = \frac{\omega_c}{S_{\omega_c}} \quad (31)$$

a pro mapování pásmové zádrže je užíván tento vztah:

$$s = \frac{B * S_{\omega_c}}{S_{\omega_c}^2 + \omega_c^2}, \quad (32)$$

kde ω_c značí u spojitých systémů kmitočety s poklesem o 3dB. Koeficient B zastupuje šířku pásma a je ekvivalentní výrazu $1/Q$, kde Q znamená jakost filtru. Tyto vztahy jsou dosazeny do rovnic (15) nebo (17). Posledním použitým přenosem je ekvalizér, jehož přenos v rovině s je dán následovně:

$$H(s) = \frac{s^2 + \left(\frac{A}{Q}\right)s + 1}{s^2 + s/(A * Q) + 1}, \quad (33)$$

kde A je hodnota zesílení. [34][33]

Po znormování jednotlivých přenosů, kde $\omega_c=1$, je možné již dosadit za S_{ω_c} vztah pro bilineární transformaci, viz rovnice (34).

Nyní je možné vyjádřit koeficienty přenosu v rovině z pomocí bilineární transformace normovaného přenosu analogového filtru 1. řádu (14).

$$\frac{1}{s+1} = \frac{1}{c \frac{1-z^{-1}}{1+z^{-1}} + 1} \Bigg/ * \frac{1+z^{-1}}{1+z^{-1}} \quad (34)$$

$$= \frac{1+z^{-1}}{c - cz^{-1} + 1 + z^{-1}} = \frac{1+z^{-1}}{c+1 + (1-c)z^{-1}} = \frac{1}{c+1} * \frac{1+z^{-1}}{1 + \frac{(1-c)}{1+c} z^{-1}} \quad (35)$$

Z rovnice (35) je zřejmé, že výsledek odpovídá požadovanému tvaru přenosu (22). Lze tedy jednoduše odvodit výpočty jednotlivých koeficientů pro první řád dolní propusti, kde:

$$\begin{aligned} g &= 1/c+1 \\ b_0 &= b_1 = 1 \\ b_2 &= 0 \\ a_0 &= 1 \\ a_1 &= (1-c)/(1+c) \\ a_2 &= 0 \end{aligned}$$

Stejným způsobem lze vyjádřit koeficienty druhého řádu, který je dán ve svém normovaném tvaru následujícím přenosem:

$$H(s) = \frac{1}{1 + \sqrt{2}s + s^2}, \quad (36)$$

kde odmocnina z čísla dvě je rovna číslu α z rovnice (16). Bilineární transformace bude tedy odvozena v obecném tvaru.

$$\frac{1}{s^2 + \alpha s + 1} = \frac{1}{\left(c \frac{1 - z^{-1}}{1 + z^{-1}}\right)^2 + \alpha \left(c \frac{1 - z^{-1}}{1 + z^{-1}}\right) + 1} \Bigg/ * \frac{1 + z^{-1}}{1 + z^{-1}} \quad (37)$$

$$= \frac{(1 + z^{-1})^2}{c^2(1 - z^{-1})^2 + \alpha c(1 - z^{-1})(1 + z^{-1}) + (1 + z^{-1})^2} \quad (38)$$

$$= \frac{1z^0 + 2z^{-1} + 1z^{-2}}{c^2(1 - z^{-1})^2 + \alpha c(1 - z^{-1})(1 + z^{-1}) + (1 + z^{-1})^2} \quad (39)$$

$$= \frac{1 + 2z^{-1} + 1z^{-2}}{c^2(1 - z^{-1})^2 + \alpha c(1 - z^{-1})(1 + z^{-1}) + (1 + z^{-1})^2} \quad (40)$$

$$= \frac{1}{1 + \alpha c + c^2} * \frac{1 + 2z^{-1} + 1z^{-2}}{1 + \frac{2(1 - c^2)}{1 + \alpha c + c^2} z^{-1} + \frac{1 - \alpha c + c^2}{1 + \alpha c + c^2} z^{-2}} \quad (41)$$

Z výsledku je možné vyjádřit koeficienty dolní propusti 2. řádu pomocí následujících vztahů:

$$\begin{aligned} g &= 1/(1 + \alpha c + c^2) \\ b_0 &= b_2 = 1 \\ b_1 &= 2b_0 \\ a_0 &= 1 \\ a_1 &= 2(1 - c^2)/(1 + \alpha c + c^2) \\ a_2 &= (1 - \alpha c + c^2)/(1 + \alpha c + c^2) \end{aligned}$$

Pomocí vypočtených vztahů je možné vyjádřit koeficienty jakéhokoli řádu pomocí bikvadratických sekcí. Výpočty jednotlivých koeficientů jsou provedeny analogicky pro jakýkoli přenos filtru.

Ověření předchozích výpočtů pomocí MATLABu

Pro ověření výpočtů bylo použito prostředí MATLAB verze 7.9.0.529 (R2009b). Ověření spočívá ve srovnání výsledků analytické metody, která je zapsána ve skriptu M-file a použitím nástroje FDATool (Filter Design and Analysis Tool). Výpočet byl proveden pro Butterworthovu dolní propust druhého řádu při vzorkovací frekvenci 48KHz a zlomovým kmitočtem 1KHz.

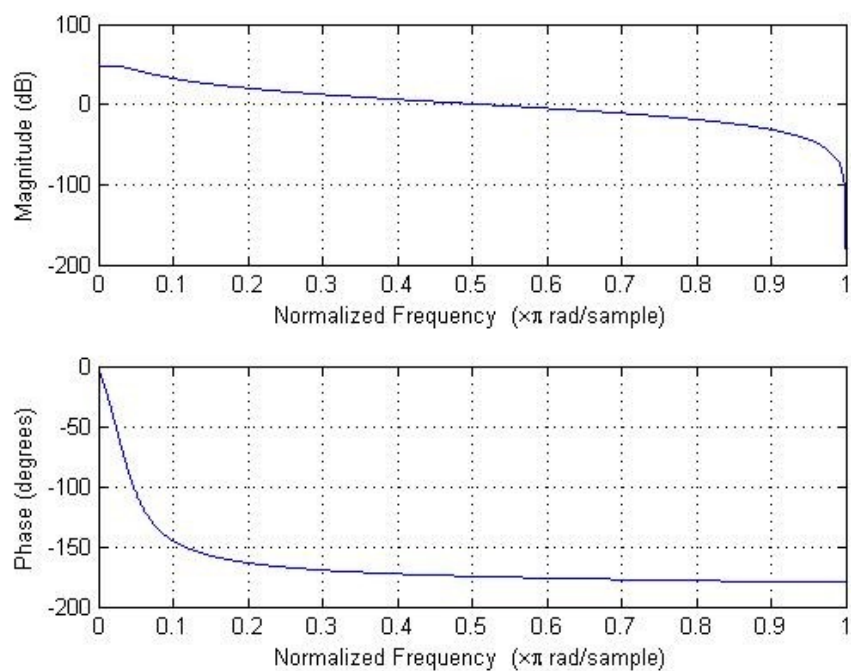
Skript M-file

Pro vykreslení frekvenční charakteristiky filtru byla užita funkce „freqz“, kde vstupními parametry je přenos číslicového filtru a počet vzorků, pro které budou počítány výsledné hodnoty. Druhou použitou funkcí je „zplane“, která vykreslí polohu nul a pólů na jednotkové kružnici.

```
clear all;
close all;
% Vzorkovací frekvence
fs = 48000;
% zlomový kmitočet
fcut = 1000;
% normálová frekvence
f=fcut/fs
% konstanta c
c=cot(pi*f)
% konstanta alfa
alfa=2*cos(pi/4)
% Koeficienty čitatele
NUM=[1 2 1]
% Koeficienty jmenovatele
A0=(c^2+alfa*c+1)
a0=1
a1=(2-2*c^2)/A0
a2=(c^2-alfa*c+1)/A0
DEN = [a0 a1 a2]
% Zesílení
g=1/A0
% Vykreslení frekvenční odezvy digitálního filtru
figure(1);
freqz(NUM,DEN,[fs])
% vykreslení rozložení nul a pólů digitálního filtru
figure(2);
zplane(NUM,DEN)
```

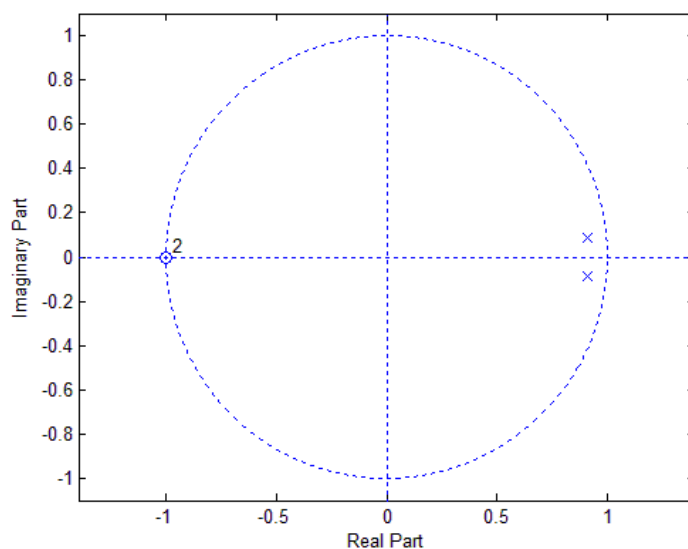
Výsledky výpočtu:

a0 = 1
a1 = -1.8153
a2 = 0.8310
g = 0.0039



obr. 27. Frekvenční charakteristika navrženého filtru.

Z amplitudově charakteristiky je zřejmé, že začínající pokles amplitudy filtru je v grafu vyneseno na ose x přibližně u hodnoty 0,02. Protože je osa x normalizována pro vzorkovací frekvenci 48kHz, je pro získání hodnoty x v Hz nutný přepočet. Hodnota x v Hz je získána vynásobením vzorkovací frekvence a odečtené hodnoty. Po výpočtu je získána hodnota x v Hz, která nabývá hodnoty 960Hz, což přibližně odpovídá návrhu filtru se zlomovou frekvencí 1kHz.

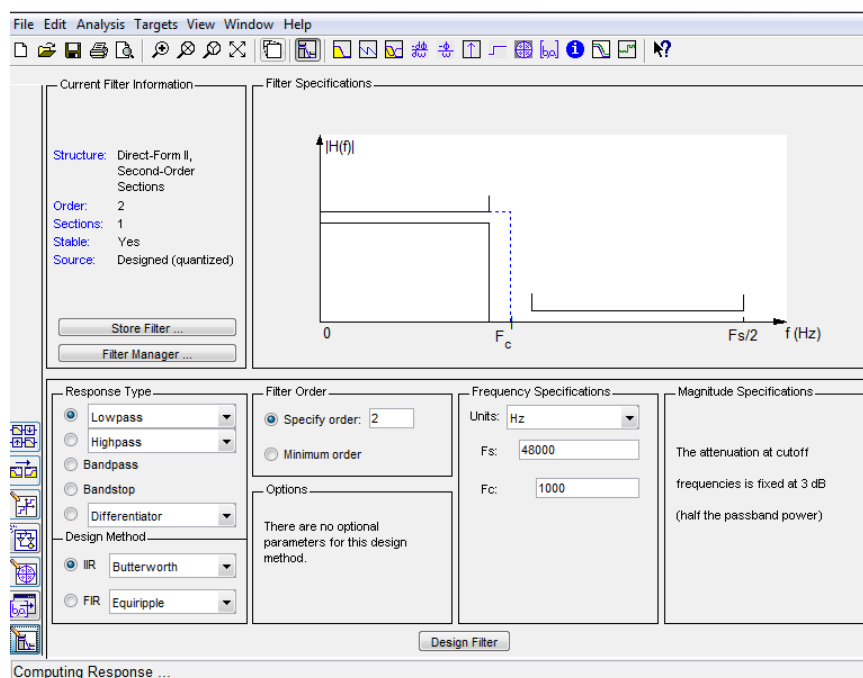


obr. 28. Zobrazení nul a pólů na jednotkové kružnici

Z rozložení nul a pólů je zřejmé, že se jedná o dolní propust s nulovým zesílením. Rozložení nul a pólů na jednotkové kružnici určuje frekvenční a amplitudové vlastnosti navrženého filtru. Polohy pólů, které jsou v grafu zobrazeny křížkem, určují frekvenční vlastnosti. Poloha nul od osy x pak určuje zesílení. V případě, že se nuly nacházejí vlevo ve směru osy x od pólů, bude se jednat vždy o dolní propust. V případě, že nuly filtru budou mít vyšší hodnotu reálné složky než póly filtru, bude se jednat o horní propust. Dalším případem je pásmová propust, ve které budou mít nuly i póly stejnou hodnotu reálné složky, avšak absolutní hodnotu imaginární části budou mít nuly nižší než póly. Posledním běžným případem je pásmová zadrž, kde umístění nul je taktéž na stejné reálné souřadnici jako v případě pásmové propusti, avšak absolutní hodnota imaginární osy bude nabývat vyšší hodnoty než u pólů filtru.

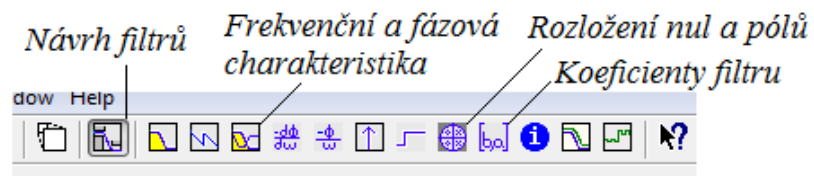
FDATool

FDATool je grafický nástroj pro návrh digitálních filtrů. Po zadání příkazu „fdatool“ do příkazového řádku MATLABu se nástroj spustí a zobrazí se následující okno, viz obr. 27.

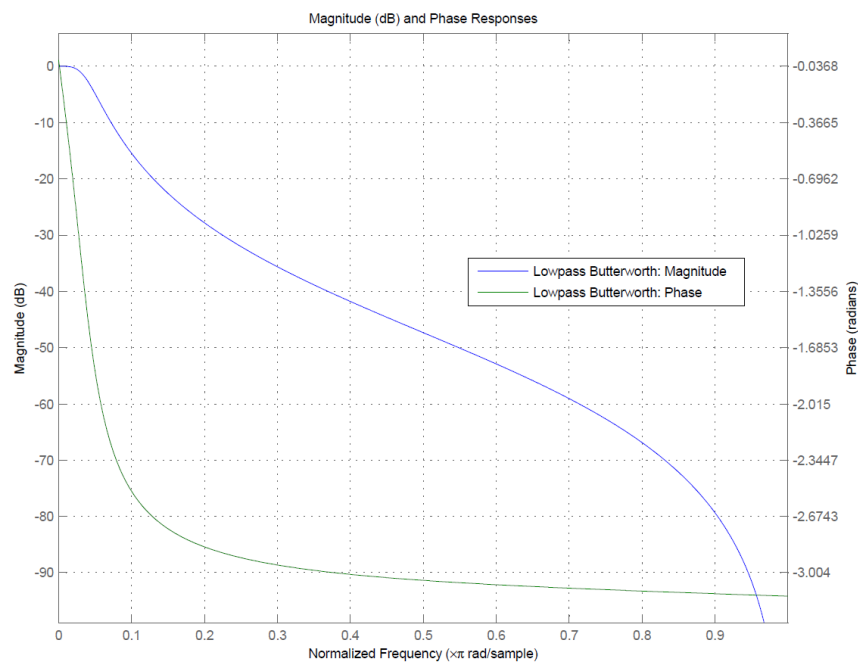


obr. 29. Nástroj FDATool

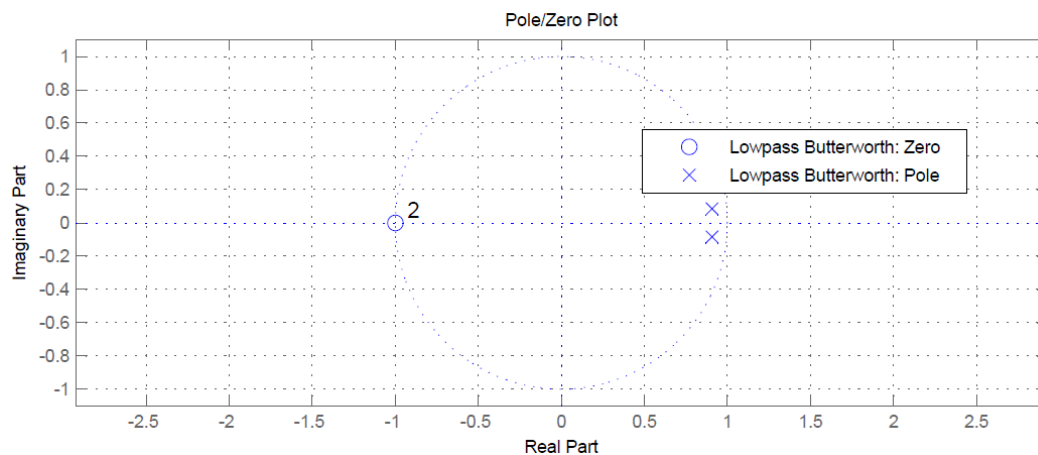
Po zadání vstupních parametrů je možné vypočítat přenos filtru a jeho charakteristiky stisknutím tlačítka „Design Filter“. Vypočtené charakteristiky je možné zobrazit pomocí ikon, které jsou umístěny pod hlavní nabídkou, viz obr. 30.



obr. 30. Ikony pro zobrazení charakteristik a výpočtů



obr. 31. Frekvenční charakteristika navrženého filtru pomocí FDAtool



obr. 32. Rozložení nul a pólů navrženého filtru.

Vypočtené koeficienty pomocí FDATool:

Numerator:

1

2

1

Denominator:

1

-1.8153410827045682

0.83100558934675761

zesílení:

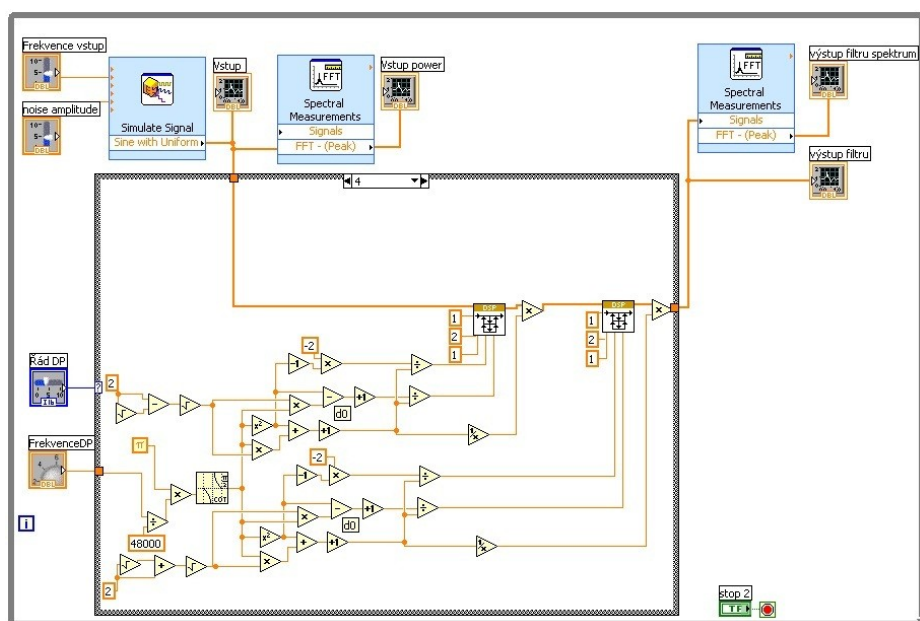
0.0039161266605473692

Z předchozích výsledků je patrné, že návrh Butterworthovy dolní propusti 2. řádu pomocí analytického výpočtu nabývá stejných hodnot jako při použití nástroje FDATool. Z rozložení nul a pólů je zřejmé, že se nacházejí na stejných souřadnicích jednotkové kružnice, a tak je možné říci, že oba přenosy budou mít stejnou charakteristiku. To potvrzují i výpočty jednotlivých koeficientů, které jsou také totožné. Vypočtené koeficienty se liší pouze v zaokrouhlení, kdy při výpočtu pomocí M-File skriptu jsou hodnoty zaokrouhledny na čtyři platná desetinná čísla.

Nyní je možné aplikovat ověřené výpočty v prostředí LabVIEW Infinity Project.

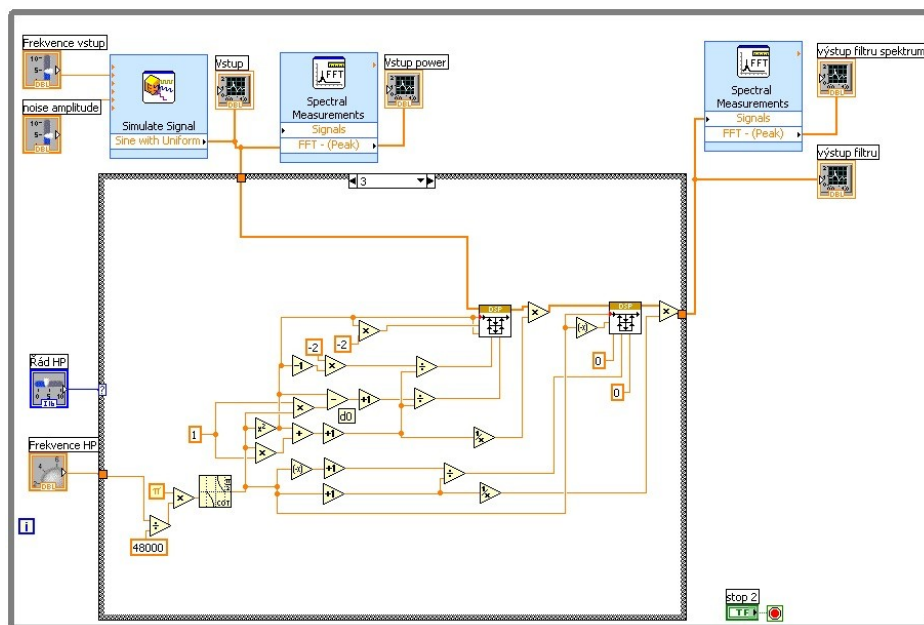
Dolní propust

Algoritmus dolní propusti je zobrazen na následujícím obr. 33. Pro testování je opět použit blok „Spectral Measurements“, jehož charakteristiky jsou opět vykreslovány na grafech hlavního panelu. Taktéž je použit blok „Simulate Signal“, na jehož vstupy jsou připojeny posuvné potenciometry hlavního panelu, kterými se mění frekvence a amplituda bílého šumu. Nejdůležitější částí algoritmu je case struktura, která přepíná jednotlivé řády filtru. V každé z jednotlivých výběrů case struktury je použit jeden či dva bikvadratické bloky filtru v závislosti na vybraném řádu filtru a výpočet koeficientů. Celý algoritmus opět běží ve smyčce while, dokud není stisknuto tlačítko „STOP“ na hlavním panelu. Na obr. 33. je zobrazen případ dolní propusti 4. řádu.



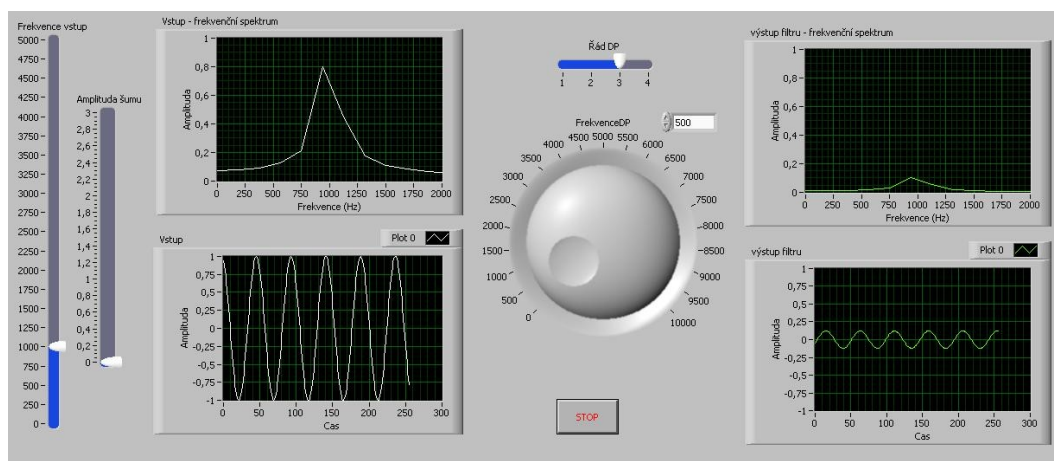
Horní propust

Algoritmus horní propusti má analogii v předchozím příkladu, odlišnosti jsou pouze ve výpočtu koeficientů b přenosu $H(z)$, viz příklad horní propusti 3. řádu, který je na obr. 34.



Hlavní panel pro horní a dolní propust

Rozložení komponentů pro monitorování a změnu parametrů je vidět na následujícím obr. 35. Na obrázku je vyobrazen panel dolní propusti. Pro horní propust je rozložení komponentů stejné jako u dolní propusti, a proto zde není uvedeno.

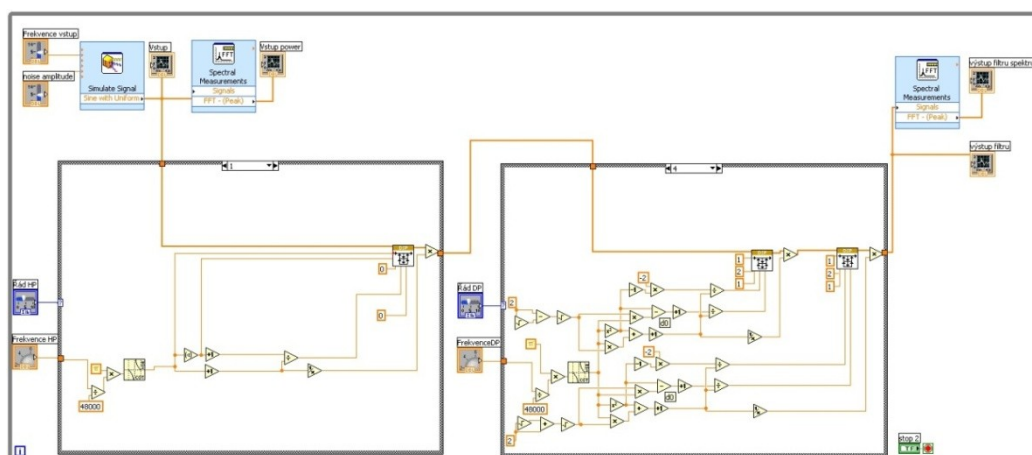


obr. 35. Hlavní panel dolní propusti

Na obr. 35 je vyobrazena filtrace vstupního signálu s kmitočtem 1kHz a nastavením dolní propusti 3. řádu se zlomovou frekvencí filtru 500Hz. Pokles amplitudy po filtraci odpovídá teoretickým předpokladům Butterworthova DP filtru 3. řádu.

Pásmová propust

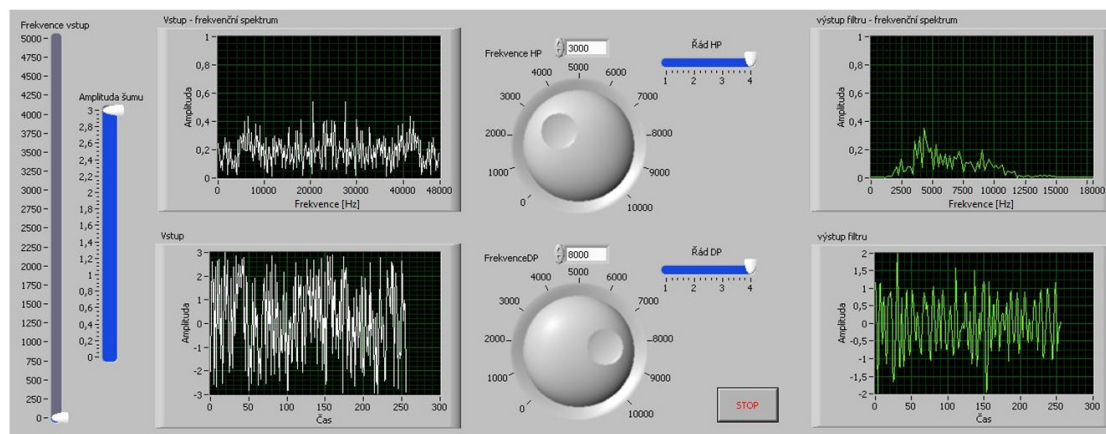
Pásmová propust je realizována kaskádním řazením horní a dolní propusti, kdy jsou použity jednotlivé case struktury z předchozích příkladů DP a HP, viz obr. 36. Zbytek kódu je analogický jako u předchozích příkladů.



obr. 36. Algoritmus pásmové propusti

Hlavní panel pásmové propusti

Hlavní panel pásmové propusti vyobrazený na obr. 37 se od předchozích příkladů liší pouze přidáním jednoho potenciometru a posuvného potenciometru pro HP/DP. HP a DP je možné nastavit separátně, což může přispět k lepší použitelnosti filtru při jeho aplikaci.

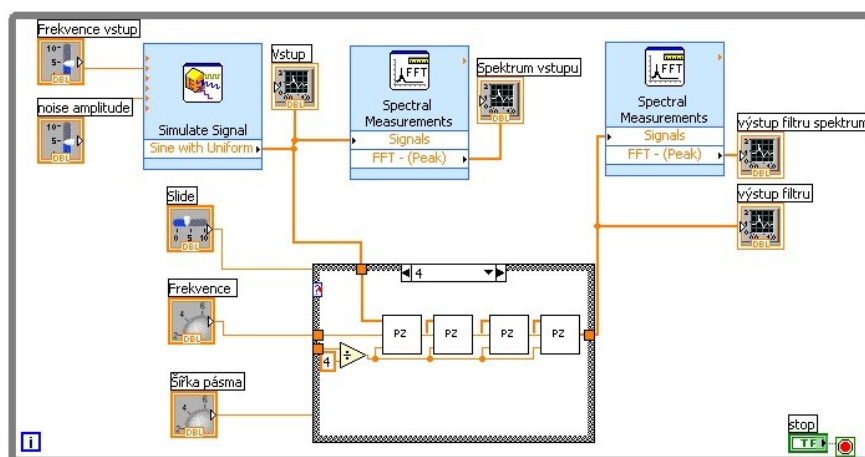


obr. 37. Hlavní panel pásmové propusti

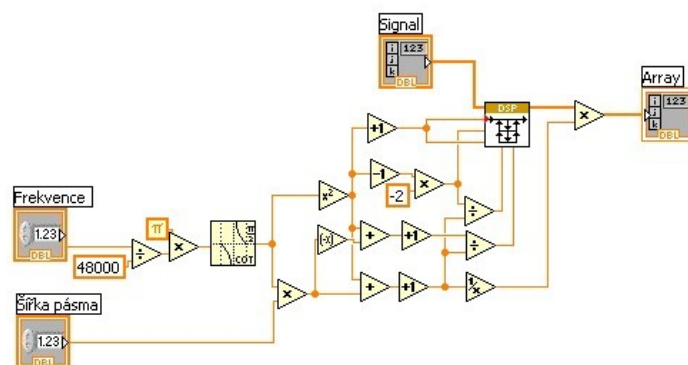
Z výstupní frekvenční charakteristiky, která je zobrazena jako oboustranné spektrum v grafu na hlavním panelu (vpravo nahoře), je zřejmá funkčnost pásmové propusti.

Pásmová zadrž

Pásmová zadrž je realizována kaskádním řazením jednotlivých SubVI, které obsahuje jeden blok bikvadratického filtru spolu s koeficienty pro výpočet. SubVI „PZ“ jsou opět vloženy do case struktury, která přepíná jednotlivé řady filtrů. Dalším nastavením je změna kmitočtu a šířky pásma. Při měření se vyskytla v kódu chyba, která je na obr. 38 již opravena. V případě měření nebyla adekvátně snížena hodnota šířky pásma. Snížení spočívá vydělením hodnoty šířky pásma číslem 1-4 podle použitého řádu. To je patrné i z naměřených charakteristik. Zbylé použité bloky mají stejnou funkci jako v předchozích příkladech.



obr. 38. Algoritmus PZ

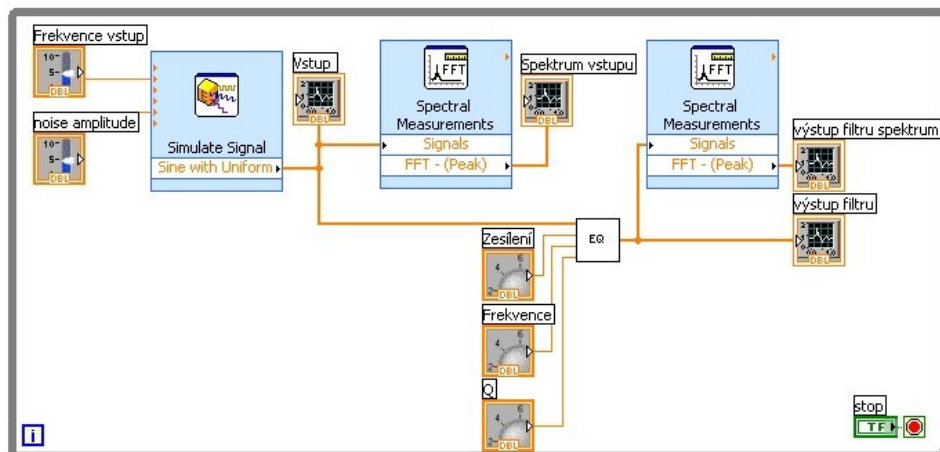


obr. 39. SubVI pásmové zádrže

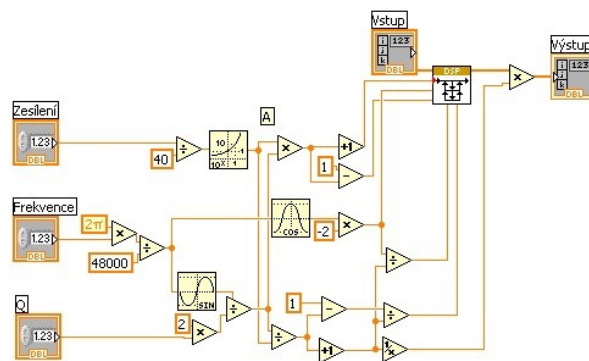
Na obr. 39 je zobrazen kód SubVI, ve kterém je umístěn blok bikvadratického filtru s příslušným výpočtem koeficientů pásmové zádrže.

Parametrický ekvalizér

Na obr. 40 je vyobrazen algoritmus parametrického ekvalizéru, u kterého je možné měnit jakost Q, frekvenci a zesílení. Z důvodů použití kódu ekvalizéru ve funkci ABL bylo vytvořeno SubVI s názvem EQ. Ostatní použité bloky mají opět stejnou funkci jako v předchozích příkladech.



obr. 40. Algoritmus parametrického ekvalizéru



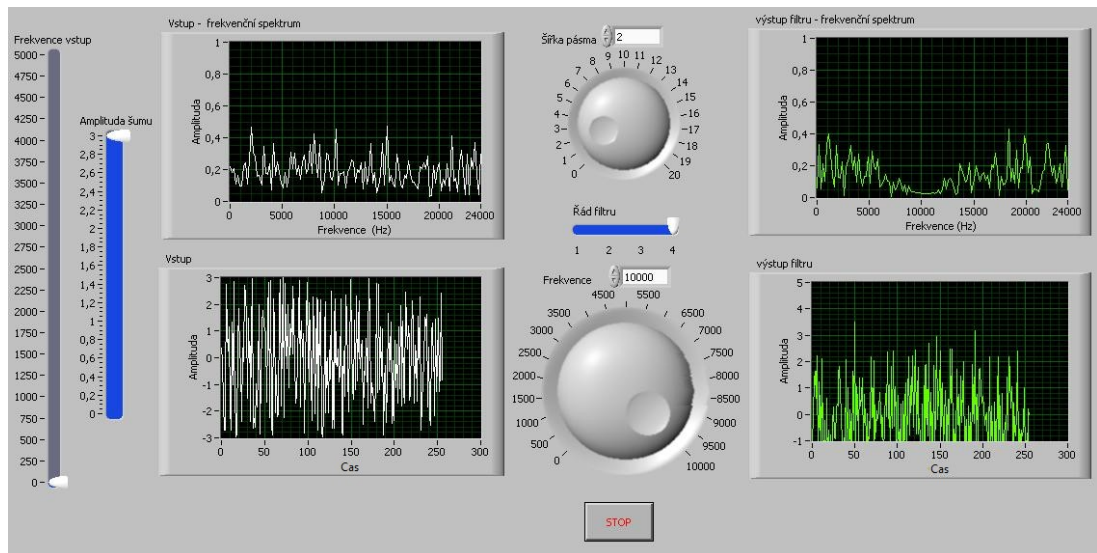
obr. 41. SubVI parametrického ekvalizéru

Na obr. 41 je vyobrazen algoritmus, který je obsažen v SubVI EQ. Hlavním blokem je opět bikvadratický filtr s výpočtem jednotlivých koeficientů.

Hlavní panel pro pásmovou zádrž a parametrický ekvalizér

Hlavní panel, jež je vyobrazen na obr. 42, má analogické rozložení ovládacích komponentů jako v předchozích příkladech. Otočnými potenciometry se mění šířka pásma a frekvence. Tahový horizontální potenciometr pak slouží ke změně řádu. Vertikální potenciometry po levé straně slouží k nastavení generovaného signálu.

Hlavní panel parametrického ekvalizéru má analogické zobrazení s rozdílem záměny horizontálního tahového potenciometru za otočný, jež reprezentuje zesílení. Šířka pásma má u ekvalizéru funkci jakosti Q .



obr. 42. Hlavní panel pásmové zádrže

4.2 Filtrace ve frekvenční oblasti

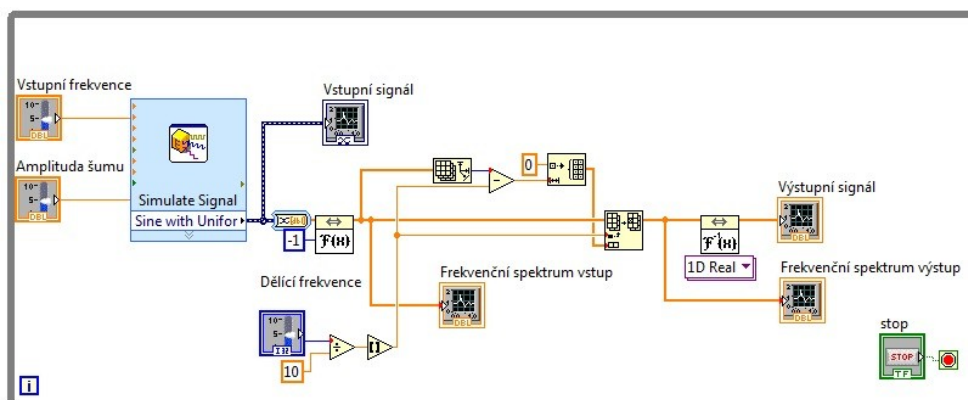
Pro filtraci ve frekvenční oblasti byla použita funkce FFT a inverzní funkce FFT pro zpětný převod filtrovaného signálu. Filtrace ve frekvenční doméně pracuje s určitým počtem vzorků popisujících jednu periodu signálu na jednotkové kružnici. Z tohoto faktu vyplývá vztah pro výpočet přírůstku frekvence jednotlivého vzorku, který je možné vyjádřit následovně:

$$\Delta f = \frac{f_s}{N}, \quad (42)$$

kde Δf značí přírůstek frekvence pro jednotlivé vzorky N při vzorkovací frekvenci f_s . Právě tento vztah je stěžejní pro následující výpočty, ale také pro nastavení grafů, které vykreslují frekvenční charakteristiky.

FFT filtrace užívající dynamické pole

Následující algoritmus, který je vyobrazen na obr. 43, je určen pro PC a jedná se tak pouze o simulaci. Kód byl vytvořen pro otestování funkce FFT filtrace a je složen z pěti hlavních bloků, kterými jsou reálná FFT, zjištění velikosti pole, generování pole, vložení pole a inverzní reálná FFT. Vstup signálu zajišťuje opět blok „Simulate Signal“, který je nastaven na vzorkovací frekvenci 44,1kHz s počtem vzorků o hodnotě 4410. Δf je tedy právě 10Hz. Generovaný signál je následně převeden do frekvenčního spektra, které reprezentuje pole, jež má právě 4410 vzorků. Z posuvného potenciometru, který určuje dělicí kmitočet, je hodnota přivedena na blok vložení pole, kde hodnota určuje pozici vložení vygenerovaného pole a dále přichází do rozdílu s počtem vzorků, což určuje velikost generovaného pole. Kód tedy funguje tak, že je vygenerováno pole „0“, které je uloženo na konkrétní pozici. Tím se přemaže původní hodnota daných vzorků. Po těchto operacích vstupuje upravené pole do zpětné FFT.

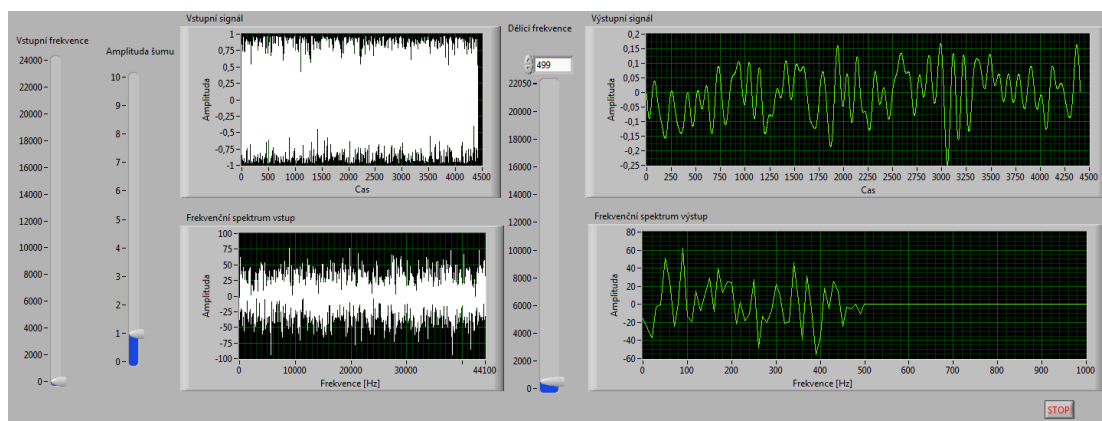


obr. 43. FFT filtrace na PC užívající dynamické pole

Tento kód byl aplikován na kitu NI Speedy – 33, avšak kit nepovolil vytvoření dynamického pole a proto byl při aplikaci nepoužitelný.

Hlavní panel FFT užívající dynamické pole

Hlavní panel této aplikace je zobrazen na obr. 44. Je tvořen čtyřmi grafy a třemi tahovými potenciometry. Potenciometry v levé části obrázku nastavují vstupní signál a prostřední potenciometr určuje dělicí frekvenci.

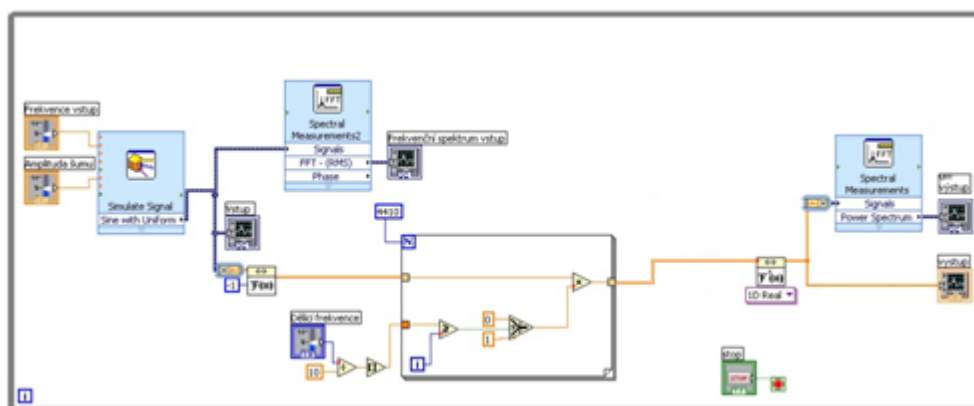


obr. 44. Hlavní panel FFT filtrace užívající dynamické pole

V příkladu zobrazeném na obr. 44 je nastavena dělicí frekvence na 500Hz, což odpovídá frekvenčnímu spektru, které se nachází v pravé – dolní části obrázku.

FFT filtrace užívající statického pole

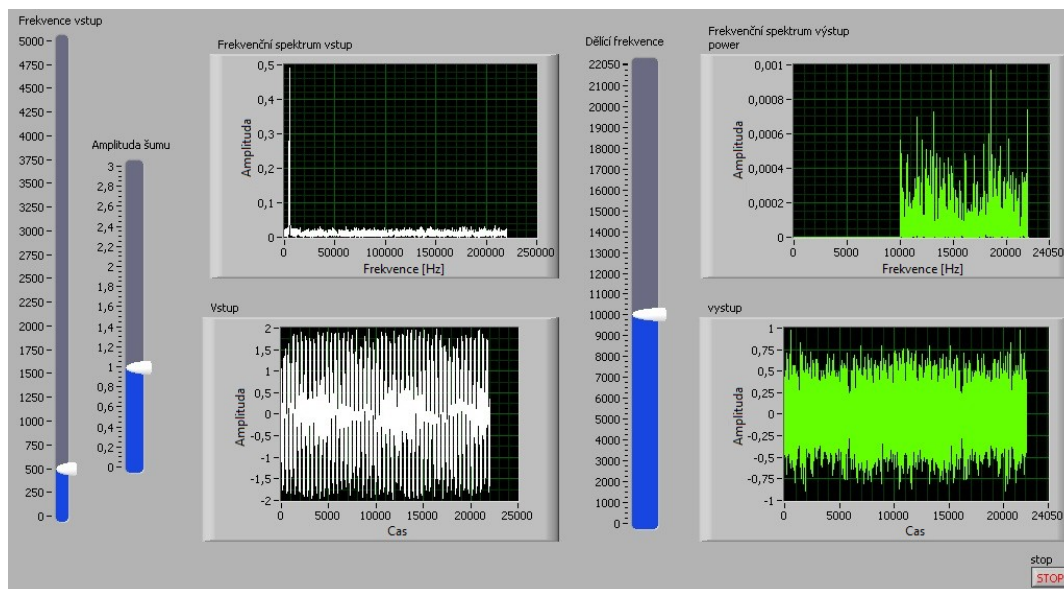
Na následujícím obr. 45 je znázorněn algoritmus užívající statického pole. Pro práci s vlastními vzorky v poli je použita smyčka for, která proběhne 4410 krát a umožňuje práci s jednotlivými vzorky. V cyklu for se nachází tři bloky – komparátor, selektor a násobení. Komparátor porovnává aktuální hodnotu proběhlých cyklů se vstupní hodnotou potenciometru. Výstup z komparátoru je binární a určuje selektoru, kterou hodnotu má poslat do bloku násobení. Tato logika zajistí buď vynulování prvků v poli nebo je ponechá beze změny. Právě tento komparátor určuje, zda se jedná o horní nebo dolní propust.



obr. 45. Algoritmus FFT filtrace

Hlavní panel FFT filtrace užívající statického pole

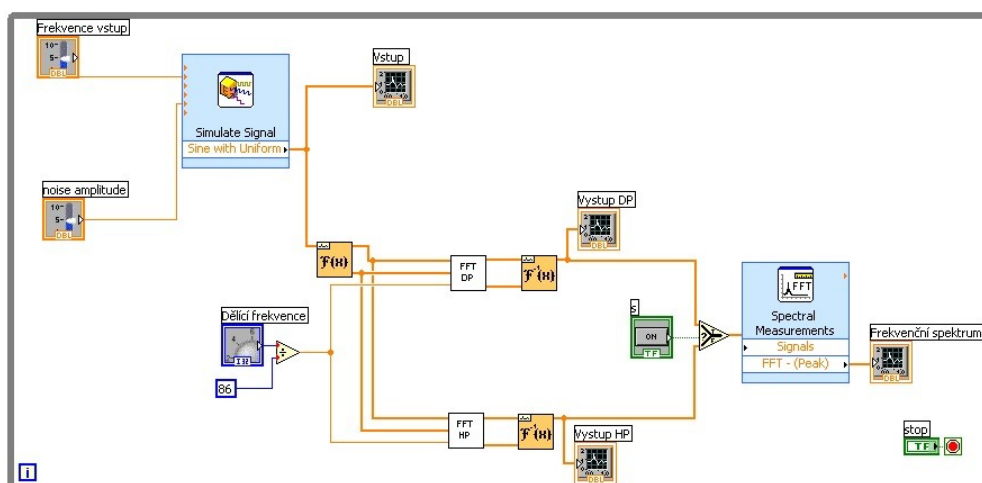
obr. 46 zobrazuje hlavní panel, který je analogický s předchozím příkladem. Dělicí frekvence horní propusti je nastavena na 10kHz, což odpovídá zobrazenému frekvenčnímu spektru.



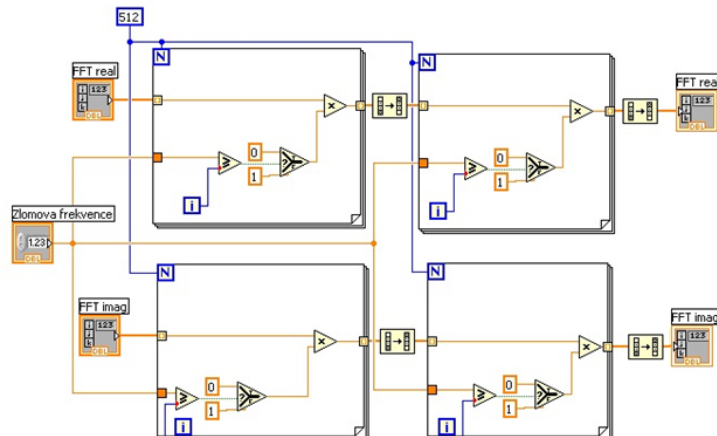
obr. 46. Hlavní panel FFT filtrace

FFT filtrace aplikovaná na kitu NI Speedy – 33

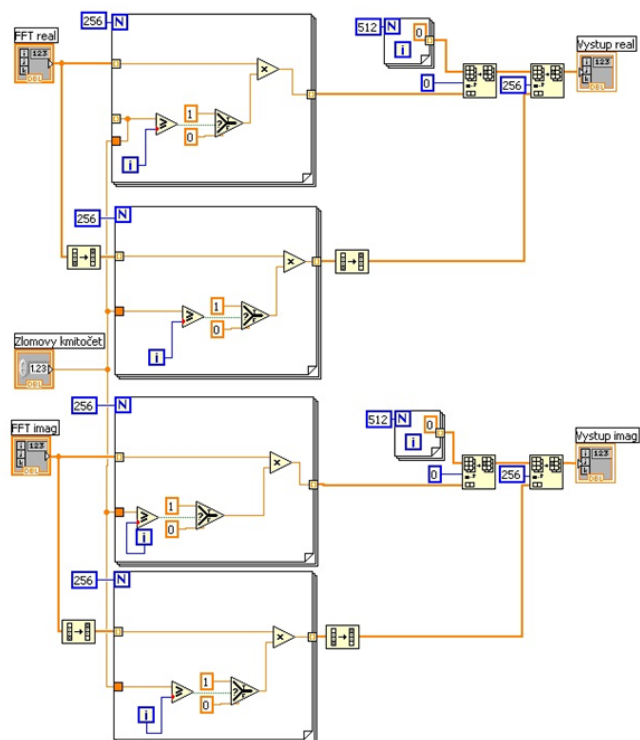
Algoritmus, jež je zobrazen na obr. 47, je založen na stejné myšlence, která byla popsána v předešlém příkladu. Protože se funkce „real FFT“ na kitu NI Speedy – 33 liší od funkce „real FFT“ při použití na PC, je potřeba patřičně upravit i algoritmus. Zatímco na PC funkce FFT vracela data pouze v jednom poli, FFT funkce obsažená v LabVIEW DSP modulu vrací pole reálných a imaginárních hodnot. Dalším rozdílem je, že funkce FFT v LabVIEW DSP modulu pracuje s oboustranným spektrem, kdežto FFT použitá na PC vrací pouze jednostranné spektrum. Z těchto důvodů je kód rozlehlejší, a proto byly vytvořeny dvě SubVI. První je navrženo jako DP, druhé pak jako HP. Z obrázku je zřejmé, že výstupy jednotlivých filtrů se dělí o jeden blok spektrální analýzy, která je vykreslována v závislosti na stlačení tlačítka z hlavního panelu. Důvodem je, že vytvořený kód byl pro HW již příliš náročný a po přidání dalšího bloku spektrální analýzy se již program nezkompiloval a byla zobrazena chybová hláška o nedostupné paměti. Proto nebyla spektrální analýza přidána ani na vstupním signálu. Ostatní použité prvky mají stejnou funkci jako v předchozích úlohách. Důležitým nastavením pro FFT filtrace je počet vzorků a vzorkovací frekvence. Oba parametry jsou zapsány v bloku Simulate Signal, kde vzorkovací frekvence byla nastavena na 44,1kHz a počet vzorků na 512. Z toho vyplývá, že rozdíl frekvence mezi jednotlivými vzorky je 86,1328125Hz. V případě vysokých frekvencí je tento krok nadbytečný, u použití basových kmitočtů je však tato posloupnost vzorků příliš hrubá pro kvalitní filtrace. Bohužel maximální počet vzorků, který DSP kit povoluje, se pohybuje okolo 750. Tento fakt však ovlivňuje četnost a složitost funkcí použitých v programu.



Na obr. 48 je zobrazen algoritmus SubVI horní propusti. Jako vstupní parametry této funkce jsou dvě pole a hodnota dělicí frekvence. Operace prováděné s reálnou a imaginární částí jsou totožné. Nejprve je z pole vynulována první polovina hodnot nízkých kmitočtů, poté se pole reverzuje a postup se opakuje s druhou stranou spektra. Následně je pole opět převráceno do původní posloupnosti prvků, avšak již s vynulovanými hodnotami příslušných nízkých kmitočtů oboustranného spektra.



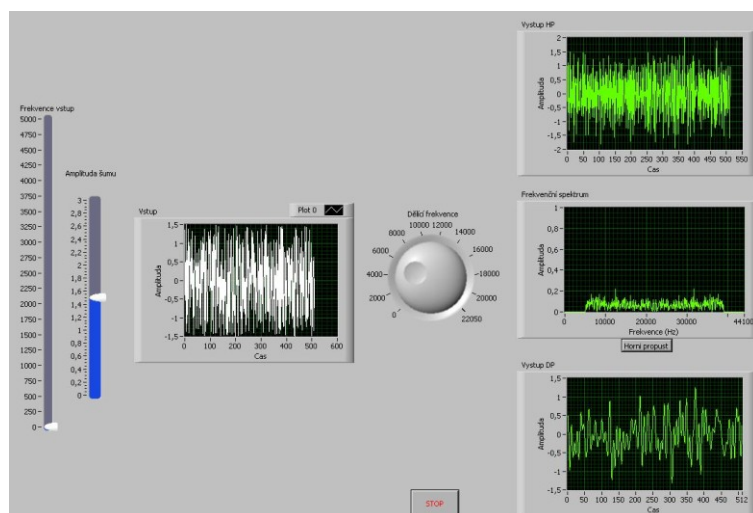
Algoritmus DP, jež je zachycen na obr. 49, je o něco složitější než v případě HP. Je to proto, že je zde třeba zvlášť pracovat s první a druhou polovinou pole. To proto, že nyní je potřeba nulovat vyšší kmitočty a v případě, že by bylo načteno celé pole, vynulovala by se z pole s vysokými kmitočty i druhá strana spektra nízkých kmitočtů. Z tohoto důvodu je pole rozděleno na dvě poloviny pomocí dvou smyček for, kde každá načte jednu polovinu původního pole. Následně jsou filtrované pole opět složeny do jednoho.



obr. 49. Sub VI FFT DP

Hlavní panel FFT filtrace aplikované na kitu NI Speedy – 33

Grafické rozhraní, které je na obr. 50, se skládá ze 4 grafů a tří potenciometrů. Prostřední graf zobrazuje oboustranné frekvenční spektrum po filtraci. Pod grafem se nachází tlačítko s popisem aktuální volby (horní nebo dolní propust).



obr. 50. Hlavní panel FFT filtrace

5 Ověření funkčnosti navržených filtrů

Tato kapitola je věnována reálné simulaci navržených filtrů a jejich ověření pomocí bílého šumu nebo harmonického signálu. Kódy jednotlivých úloh jsou vždy spuštěny na kitu NI Speedy – 33, avšak nejsou připojeny fyzické vstupy/výstupy. Vstupní signál je generován pomocí bloku „Simulate Signal“, který byl zobrazen v předchozích kódech jednotlivých návrhů filtrů (např. obr. 47). Filtrovaný signál je následně přiveden na bloky grafu a „Spectral Measuremens“, který následně vykreslí do dalšího grafu oboustranné frekvenční spektrum. Výstupní průběhy vykreslené v grafech na hlavním panelu jsou pro ověření funkčnosti navržených filtrů stěžejní.

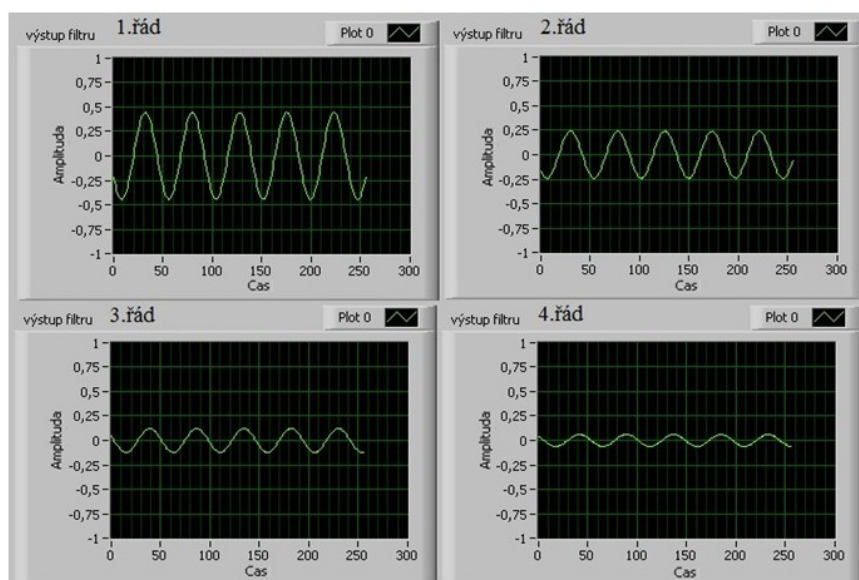
Následující podkapitoly popisují jednotlivé metody ověření, které byly při práci použity.

5.1 Ověření dolní, horní a pásmové propusti

V případě filtrů DP, HP a PP byl pro testování použit harmonický signál o frekvenci 1kHz při vzorkovací frekvenci 48kHz s 256 vzorky. Čas udávaný na osách x grafů je zobrazen v časové posloupnosti vzorků. Z průběhu harmonického signálu na obr. 51 je zřejmé, že jedna perioda trvá přibližně 50 vzorků, pro které odpovídá následující vztah:

$$50 * \frac{1}{48 * 10^3} \approx 1,0417ms, \quad (43)$$

kde 48000 značí vzorkovací frekvenci. Z výsledku je zřejmé, že jedna perioda signálu trvá přibližně 1ms, což odpovídá vstupnímu kmitočtu 1kHz. Tento výpočet se vztahuje ke všem případům užitých v diplomové práci. Různí se pouze počet vzorků, případně vzorkovací kmitočet.



obr. 51. Pokles amplitudy jednotlivých řádů filtrů na dvojnásobné frekvenci

Na obr. 51 lze spatřit pokles amplitudy u jednotlivých řádů filtrů. V případě dolní propusti byla zlomová frekvence nastavena na 500Hz, což odpovídá polovině vstupního kmitočtu. Na základě těchto faktů je možné srovnat útlum jednotlivých řádů filtrů s teoretickými hodnotami poklesu 6dB, 12dB, 18dB a 24dB pro jednotlivé poklesy řádů filtrů (1.- 4. řádu). Z následujícího vztahu (44) je možné vypočítat útlum v dB pro jednotlivé řády filtru. Pro první řád je útlum vypočítán takto:

$$20 * \log \frac{1}{1/2} = 6,02 \text{ [dB]}, \quad (44)$$

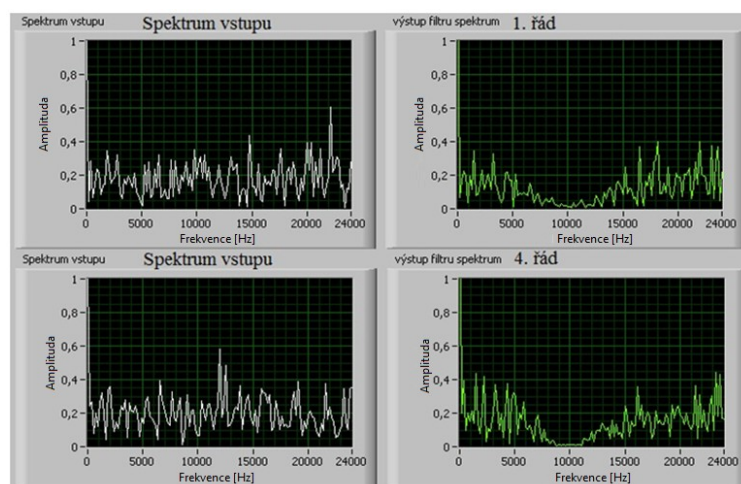
kde v logaritmovaném výrazu se počítá se vstupní amplitudou jedna a výstupní amplitudou o přibližné hodnotě jedné poloviny. Pokles 6dB na oktávu (oktáva je definována jako dvojnásobný kmitočet) odpovídá teoretickým předpokladům. Pokles amplitudy na zlomovém kmitočtu pro Butterworthův filtr je dle tabulek $1/\sqrt{2}$, což se rovná přibližně 0,707. Tento pokles byl zaznamenán pro jednotlivé řády filtrů, avšak v práci není uveden, jelikož pokles amplitudy na oktávu jednotlivých řádů filtrů je pro demonstraci dostatečný.

Stejného zobrazení jako je na obr. 51 bylo docíleno při použití HP. V případě horní propusti byl použit opět harmonický signál o kmitočtu 1kHz, zlomová frekvence byla dvojnásobná, tedy 2kHz. Tímto se docílilo stejného poklesu amplitud jednotlivých řádů filtrů a zobrazení křivek je totožné s vyobrazeným poklesem řádu DP. Proto zde zobrazení HP není uvedeno.

Testování pásmové propusti nebylo provedeno, protože návrh PP se skládá z kaskádního řazení HP a DP, které byly výše zmíněným postupem otestovány.

5.2 Ověření pásmové zádrže

Pro ověření pásmové zádrže se již harmonický signál příliš nehodí. Je to proto, že v pásmové propusti jsou použity 2 zlomové frekvence, a proto by mohl jeden zlom ovlivňovat druhý např. v případě malé šířky zadržovaného pásma. Proto byl jako vstupní signál použit bílý šum o amplitudě 3, jež byl opět generován blokem Simulate Signal. Na obr. 52 lze vidět průběh spektrální analýzy pro případ PZ 1. a 4. řádu, jejichž středová frekvence byla nastavena na 10kHz a šířka pásma B byla rovna 2. Číslo B je v tomto případě ekvivalentní výrazu $1/Q$.

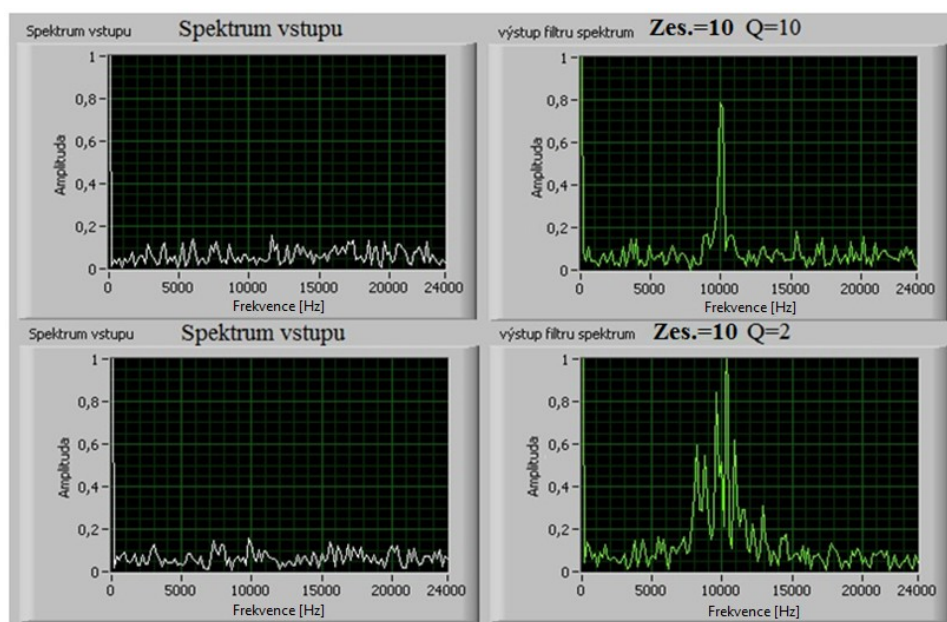


obr. 52. Frekvenční spektrum vstupu a výstupu PZ pro 1. a 4. řád

Z výsledků na obr. 52 je patrná funkčnost PZ a lze rozpoznat i strmější útlum mezi 1. a 4. řádem.

5.3 Ověření parametrického ekvalizéru

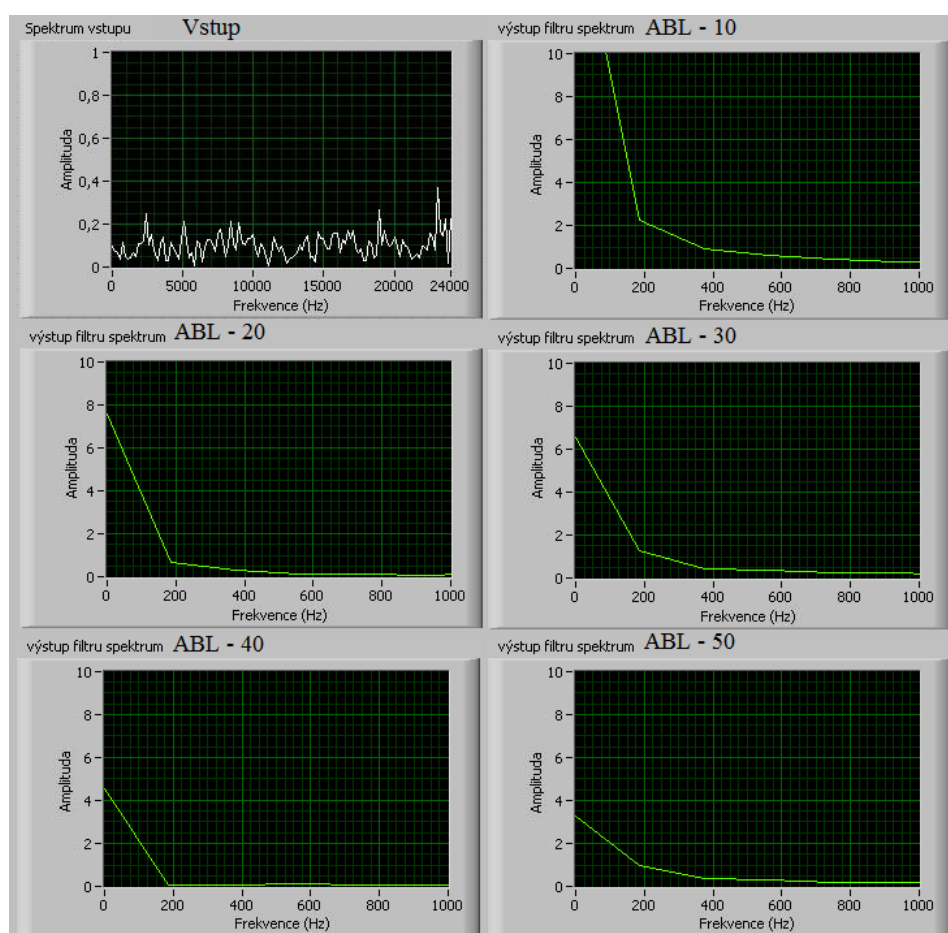
Parametrický ekvalizér byl ověřen stejnou metodou jako PZ. Jako vstupní signál byl použit opět bílý šum, tentokrát s amplitudou 1, který generoval blok Simulate Signal. Středová frekvence byla nastavena na 10kHz, zesílení bylo nastaveno na 10dB s nastavením jakosti 10 a 2. Z průběhů na obr. 53 je patrná funkčnost parametrického ekvalizéru. Hlavním parametrem, který je zde porovnáván, je hodnota Q , pro kterou je možné říci, že čím nižší bude jakost Q , tím bude větší šířka ekvalizovaného pásma.



obr. 53. Zdvih ekvalizovaného pásma signálu pro 2 hodnoty Q

5.4 Ověření funkce ABL

Pro ověření funkce ABL byl opět použit jako vstupní signál bílý šum o amplitudě 1, jež byl generován blokem Simulate Signal. Vzhledem ke kolísající amplitudě bílého šumu na jednotlivých frekvencích a poměrně velkému zesílení basových kmitočtů lze považovat simulované výsledky pouze jako orientační. To proto, že i výstupní charakteristiky byly patřičně rozkmitané v závislosti na vstupní amplitudě signálu. Z obr. 54 je však zřejmý rozdíl mezi hodnotou 10 phonů a hodnotou 50 phonů (nižší zesílení než v případě 10 phonů) funkce ABL. Hodnota 1 fonu je stanovena jako 1dB při frekvenci 1kHz. Příslušná hodnota zesílení pro basové pásmo kmitočtů je popsána v kapitole 0, ve které je blíže popsána problematika návrhu funkce ABL.



obr. 54. Zdvih basového spektra signálu pro hladinu 10-50 phonů

6 Implementace funkce ABL

Funkce ABL je založena na nelinearitách lidského sluchu, kterou se snaží v oblasti basových kmitočtů potlačit. Lidský sluch je různě citlivý v závislosti na frekvenci, viz obr. 23 v podkapitole 3.4. Nelinearita lidského sluchu je samozřejmě závislá i na jedinci, jelikož každý má odlišnou fyziologii, a tím i jinou citlivost sluchu. Citlivost sluchu se také mění s věkem jedince.

Při řešení ABL funkce byla použita norma ISO 226, která vychází z Robinson-Dadsonových křivek stejné hlasitosti (obdoba Fletcher – Munsonových křivek stejné hlasitosti, avšak křivky byly popsány o 23let později). V roce 1986 byla vydána první verze normy ISO 226, jež užívala přímo Robinson-Dadsonových křivek. Roku 2003 byla norma změněna na základě dvanácti mezinárodních studií týkající se křivek stejné hlasitosti. Dnešní norma ISO 226 popisuje křivky stejné hlasitosti pro 29 kmitočtů v rozmezí 20Hz až 12,5kHz a hladinu 0-90 phonů. Tato norma je zavedena v ČSN EN ISO 266 (01 1601) Akustika - Vyvolené kmitočty. V normě ISO 226 je popsán vztah pro výpočet akustického tlaku pro vstupní hodnotu ve phonech (1 phon je roven 1 dB při kmitočtu 1kHz). Tento vztah je použit ve funkci ABL pro výpočet zesílení jednotlivých kmitočtů a má následující znění:

Hladina akustického tlaku L_p pro čistý tón s frekvencí f , jež má hladinu L_n , je dán vztahem:

$$L_p = \left(\frac{10}{\lambda_f} \lg A_f \right) dB - L_U + 94dB, \quad (45)$$

kde

$$A_f = 4,47 * 10^{-3} * (10^{0,025L_n} - 1,15) + \left[0,4 * 10^{\left(\frac{T_f + L_U}{10} - 9 \right)} \right]^{\alpha_f}, \quad (46)$$

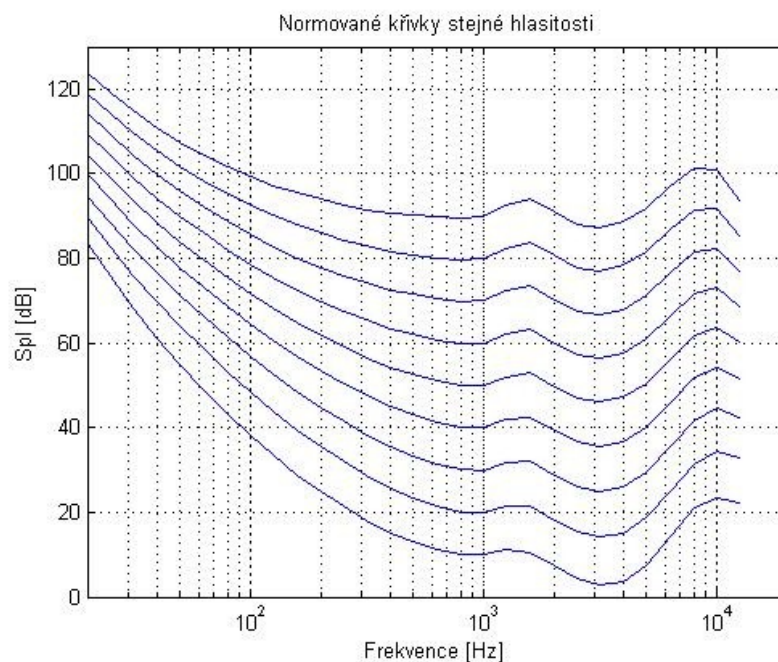
L_U značí velikost lineární přenosové funkce normované na 1kHz, α_f je exponent pro vnímání hlasitosti a T_f je práh slyšení [40]. Jednotlivé hodnoty, ze kterých je výpočet pro hladinu phonů vyjádřen, jsou uloženy v polích o 29 hodnotách.

Pro normu ISO 226 byla vytvořena funkce pro práci v MATLABu, která je dostupná na [19]. Vstupem funkce nesoucí název iso226 je hladina ve phonech a výstupem je pole hodnot akustické hladiny spjaté s frekvencí. Funkce byla použita v následujících výpočtech.

Skript M-File

```
close all;
clear all;
p=0;
for i = 1:9;
    p=p+10;
    if ((p < 0) | (p > 90))
        [spl,freq]=iso226(p);
        semilogx(freq,spl)
        hold on
        grid on
    else
        [spl,freq]=iso226(p);
        semilogx(freq,spl)
        axis([20 20000 0 130])
        title('Normované křivky stejné hlasitosti')
        xlabel('Frekvence [Hz]')
        ylabel('Spl [dB]')
        hold on
        grid on
    end
end
```

Výše napsaný kód m.file souboru slouží pro vykreslení jednotlivých charakteristik normy ISO226 (viz obr. 55) v rozmezí 10 až 90 phonů.



obr. 55. Křivky stejné hlasitosti

6.1 Simulace funkce ABL v prostředí MATLAB

Funkce ABL má za úkol zesilovat basové kmitočty v případě snížené akustické hladiny signálu. Proto byla vybrána frekvenční oblast v rozmezí 20 až 315 Hz, kde rozdíl jednotlivých akustických hladin tlaku na frekvenci 300 Hz je již poměrně lineární. Dalším důvodem výběru 300 Hz je, že kmitočty do 300 Hz ucho nevnímá směrově [18].

ABL byla řešena filtrováním jednotlivých pásem signálu s následným zesílením, jež se pohybovaly na frekvencích příslušné hodnoty převzaté z ISO226. Vymezenou oblast popisuje 13 frekvencí, a proto je zapotřebí třinácti pásmových propustí nebo ekvalizéru. Bohužel kit NI Speedy – 33 zvládl zpracovat maximálně devět PP nebo EQ. Z důvodů zachování určité výpočetní rezervy kitu bylo použito 8 ekvalizérů. Proto bylo potřeba vypočíst ze třinácti popisujících frekvencí pouze osm.

Následující kód vypočte metodou interpolace požadované hodnoty pro jednotlivé frekvence. Interpolace je realizována funkcí „spline“, která je běžně dostupná v MATLABu.

Skript M-File

```
close all;
clear all;
f = [20 25 31.5 40 50 63 80 100 125 160 200 250 315];

af = [0.532 0.506 0.48 0.455 0.432 0.409 0.387 0.367 0.349 0.33 0.315
0.301 0.288];

Lu = [-31.6 -27.2 -23.0 -19.1 -15.9 -13.0 -10.3 -8.1 -6.2 -4.5 -3.1
-2.0 -1.1];

Tf = [ 78.5 68.7 59.5 51.1 44.0 37.5 31.5 26.5 22.1 17.9
14.4 11.4 8.6];

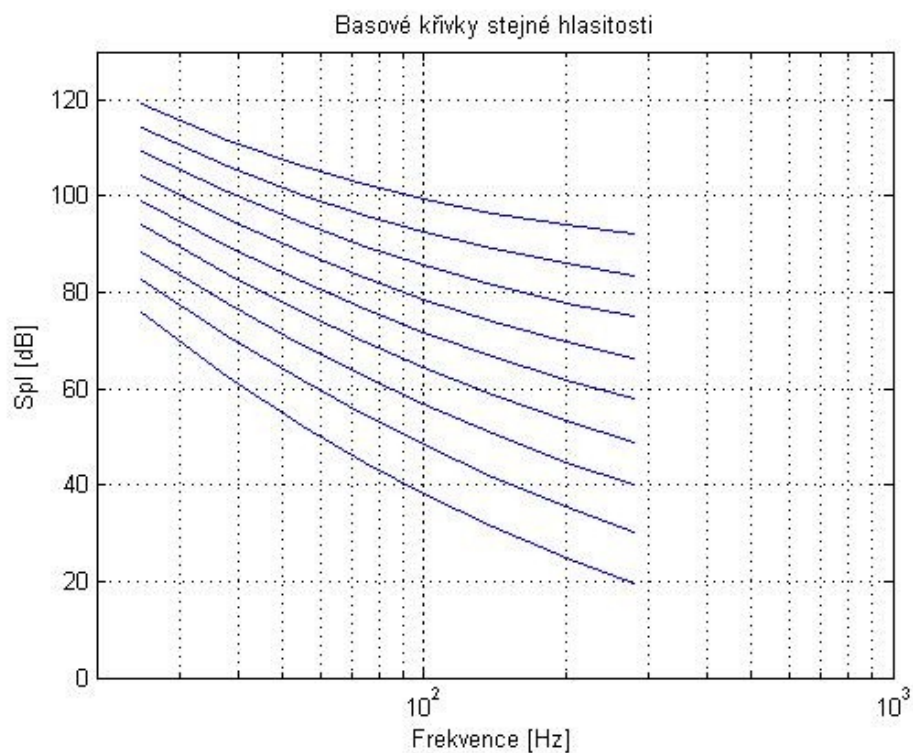
pozad_f = [25 38 55 75 100 140 200 280];

af2 = spline(f,af,pozad_f)
Lu2 = spline(f,Lu,pozad_f)
Tf2 = spline(f,Tf,pozad_f)
```

Výsledky interpolace

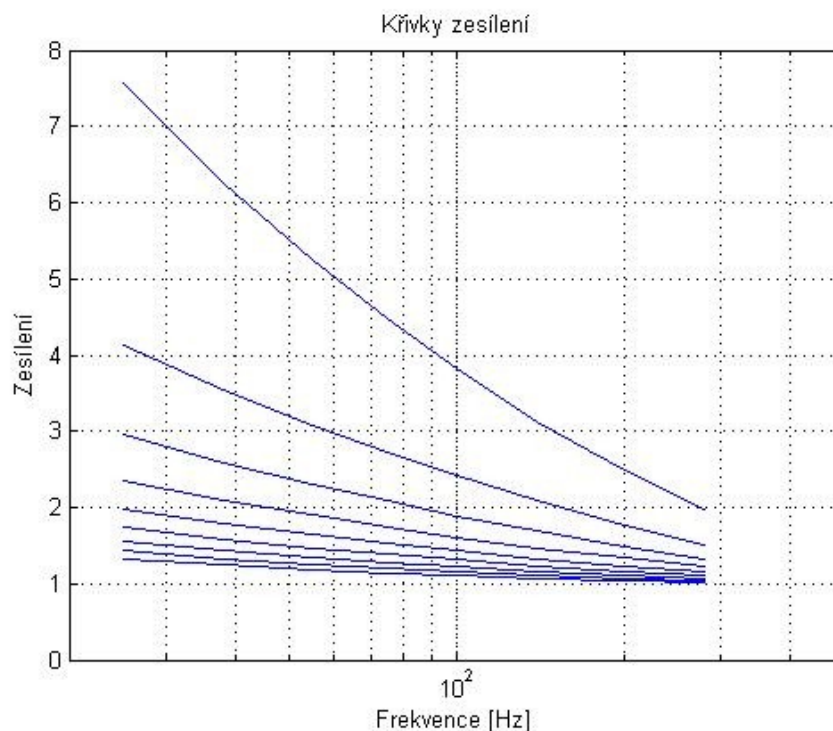
```
af2 = 0.506 0.4603 0.4223 0.3928 0.3670 0.3401 0.315 0.2943
Lu2 = -27.2 -19.8967 -14.6591 -10.9982 -8.1 -5.3759 -3.1 -1.561
Tf2 = 68.7 52.8218 41.2121 33.0552 26.50 20.0999 14.4 10.0506
```

Pro názornost byly vykresleny interpolované charakteristiky (obr. 56), jež využívají upravenou funkci „iso226“, která byla přetvořena pouze pro osm kmitočtů a nese název „iso226B“.



obr. 56. Basové křivky stejné hlasitosti po interpolaci

Pro názornost byly vykresleny jednotlivé zesilovací křivky funkce ABL (obr. 57), které používají stejný algoritmus jako v předchozím kódu, avšak je využita upravená funkce „iso226B“ a jednotlivé hodnoty Spl jsou vyděleny vstupní hladinou phonů (tedy hladinou Spl vztaženou na 1kHz).



obr. 57. Zesilovací křivky navržené funkce ABL

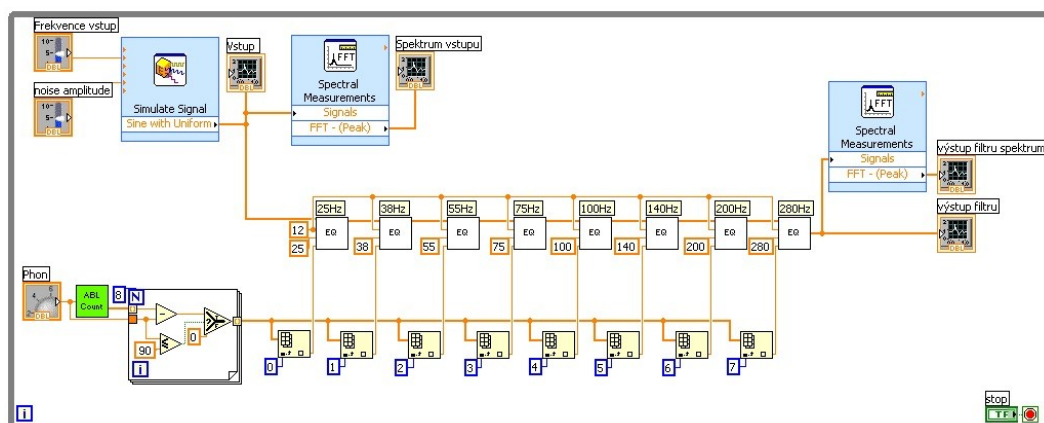
6.2 Návrh algoritmu funkce ABL

Algoritmus funkce ABL je založen na vztahu (45), který je popsán normou ISO 226. Samotný výpočet je proveden ve vytvořeném SubVI, jež nese název „ABL Count“ a je zobrazen na obr. 59. Výpočet pracuje se vstupní hodnotou, která je přivedena z potenciometru „Phon“ a třemi hodnotami, jež jsou uloženy v konstantních polích. Smyčka for, ve které je vypočítán vztah pro odvození hladiny akustického tlaku, proběhne právě 8 krát, aby byly použity všechny prvky pro výpočet. Výstupem smyčky je pole s osmi prvky, které jsou následně přivedeny do další smyčky for, kde proběhnou jednotlivé výpočty zesílení daných frekvencí. Zesílení je vypočteno v dB odečtením hodnoty ABL Count a nastavené hodnoty ve phonech. Dále je v tomto cyklu for podmínka, která zajistí použití pouze v rozsahu 0 až 90 phonů. Pokud není tato podmínka dodržena, je zesílení pro všechny frekvence nulové. Tím je zajištěn rozsah použití, který definuje použitá norma ISO 226. Výstupem ze smyčky for, jež počítá zesílení, je opět pole s osmi prvky. Následně jsou jednotlivé prvky pole vyčteny a přivedeny na konektor bloku EQ.

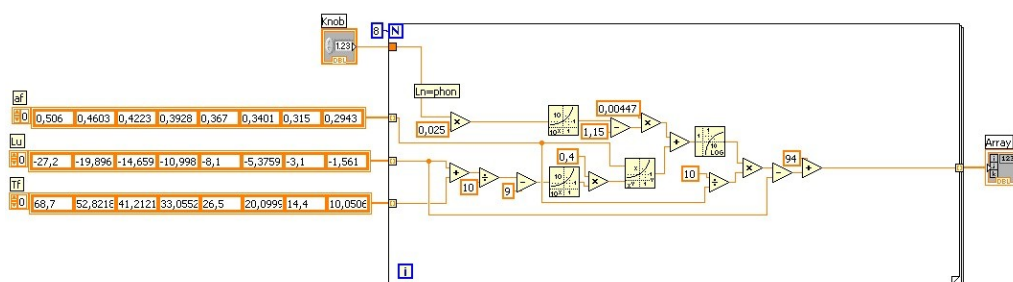
Kromě zesílení je na blok EQ přiveden vstupní signál, středová frekvence a jakost Q.

V kódu, se kterým bylo realizováno měření, byla objevena chyba. Kód byl založen na totožné myšlence, avšak pro filtraci jednotlivých kmitočtů bylo využito osmi pásmových propustí, které byly řazeny paralelně. Filtrované výstupy byly následně posčítány. Tato metoda byla chybná, jelikož při použití jednotlivých PP docházelo k fázovému posuvu. Proto jsou naměřené charakteristiky kmitajícího charakteru. Myšlenka použití PP jako EQ byla použita v [20].

Tato úloha pouze modeluje pouze zesílení, které je zadáváno z potenciometru phonů. Pro úplnost použití této funkce by kód měl být doplněn ještě o jeden potenciometr, jež by byl svázán s vyzářeným výkonem audio aparatury a potenciometrem „Phon“ tak, aby náležitě odpovídala reprodukováaná akustická hladina.



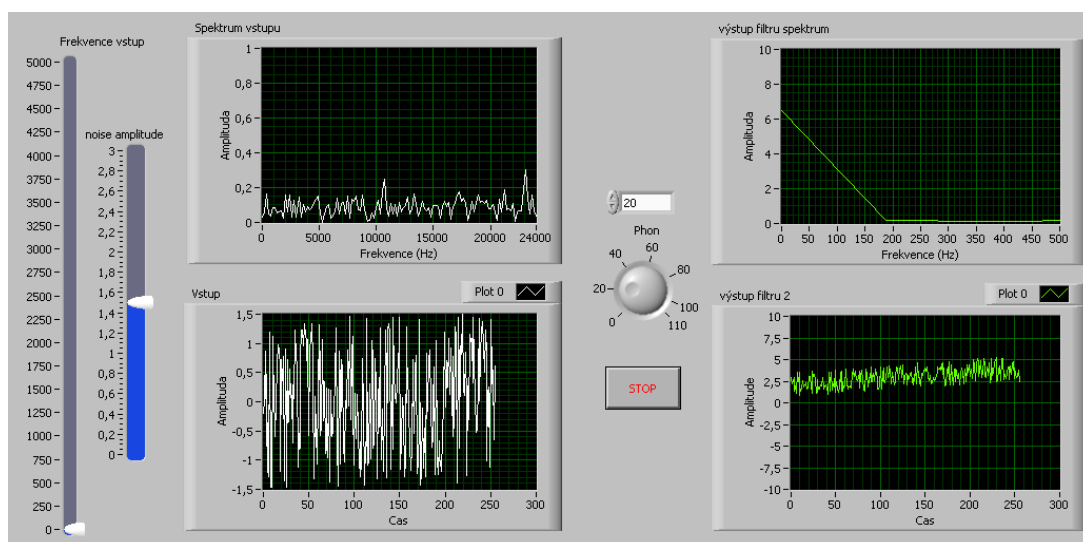
obr. 58. Algoritmus funkce ABL



obr. 59. SubVI výpočtu zesílení

6.3 Hlavní panel funkce ABL

Hlavní panel funkce ABL používá stejných konvencí a prvků jako je použito v předchozích příkladech, viz obr. 60. Hlavním ovládacím prvkem je otočný potenciometr, kterým se nastavuje požadovaná úroveň ve phonech. Další dva posuvné potenciometry jsou přiděleny generátoru signálu. Samozřejmostí je vykreslování vstupních a výstupních charakteristik ve čtyřech grafech.



obr. 60. Panel funkce ABL

7 Kit EVAL AD1940AZ

Pro srovnání kitu NI Speedy -33 zde bude stručně popsán kit EVAL AD1940AZ, na kterém byly zrealizovány a proměřeny obdobné úlohy, které jsou obsahem zadání.

EVAL AD1940AZ je vývojový kit firmy Analog Devices, jež je osazen DSP AD1940 s dvěma převodníky A/D a D/A. AD1940 je 28 bit multikanálový DSP procesor určený pro zpracování audio signálu. Procesor má 8/16 digitálních In/Out v závislosti na nastaveném módu digitální sériové sběrnice (I2S,TDM nebo jejich kombinace). DSP má v sobě integrovanou paměť RAM o velikosti 1024x28bit pro parametry a 1536x40bit pro program. Výpočetní výkon AD1940 je udáván jako 1536cyklů za jeden vzorek. Při vzorkovacím kmitočtu 192kHz jedna perioda trvá 3,39ns. Během této doby procesor vynásobí dvě 28bit čísla, přičte je do akumulátoru a vypočítá adresy následujících operandů. Z délky jednoho cyklu je možné vyjádřit takt procesoru, který je přibližně 295MHz.

AD1940 patří do rodiny „Sigma“ a je programovatelný přes komunikační sběrnici SPI pomocí stejnojmenného GUI vývojového prostředí „SigmaStudio“, kterým je možné konfigurovat podporované čipy jako jsou DSP, A/D, D/A či paměť EPROM.

Vlastní IDE je rozvrženo ve dvou oknech, viz obr. 61, kde v pravém je konfigurován HW nebo programovaná aplikace, v levém se pak nachází nabídka HW nebo programovací bloky.



obr. 61. Část zobrazení IDE SigmaStudia

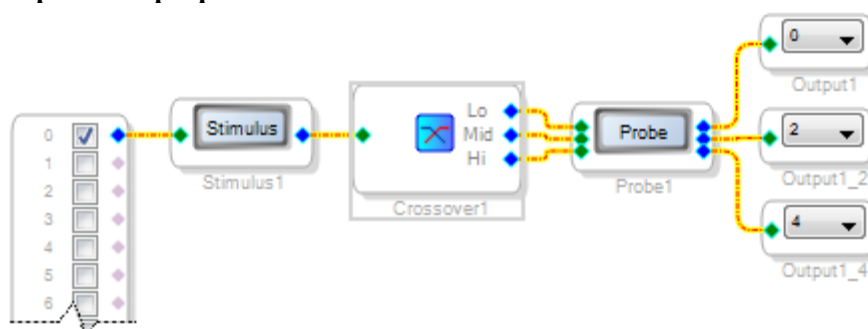
Grafické rozhraní je určeno pouze pro návrh jednotlivých aplikací a není jej možné zkompileovat. Slouží tedy pouze pro programování. SigmaStudio je založeno na myšlence ušetření času vývojářům a poskytnout tak prostor pro návrh dalších částí výsledného produktu.

Výrobce u popisu jednotlivých kitů pracujících s DSP Sigma vždy uvádí rozkreslené schéma zapojení. V příloze je proto uveden návrh schématu pro DSP ADAU144x, podle něhož bude vyhotovena deska plošných spojů.

7.1 Úlohy realizované na AD1940

Na kitu AD1940AZ bylo zrealizováno celkem pět úloh: DP,HP, PP, parametrický ekvalizér a funkce obdoba ABL. Pro příklad jsou zde zobrazeny algoritmy pracující pouze s jedním vstupním kanálem, viz obr. 62.

Dolní, horní a pásmová propust



obr. 62. Algoritmus Dolní, horní a pásmové propusti

Na obr. 62 je vyobrazen algoritmus 3 pásmové frekvenční výhybky, která se skládá z pěti typů bloků, kde první blok představuje jednotlivé digitální sériové vstupy. Stimulus a Probe jsou bloky pro zjištění přenosu filtrů, který je mezi nimi vřazen, Crossover představuje vlastní blok frekvenční výhybky a posledním jsou sériové výstupy.

Pro zobrazení simulované přenosové charakteristiky je nejdříve nutné stisknout tlačítko „Probe“, tím se zobrazí nové okno a následně stisknout tlačítko „Stimulus“, které vykreslí charakteristiku do otevřeného okna.

Nastavení výhybky je provedeno kliknutím na ikonu uprostřed bloku. Poté se zobrazí okno s nastavením, které je obdobné jako u parametrického ekvalizéru, viz obr. 64, a proto zde nebude uvedeno.

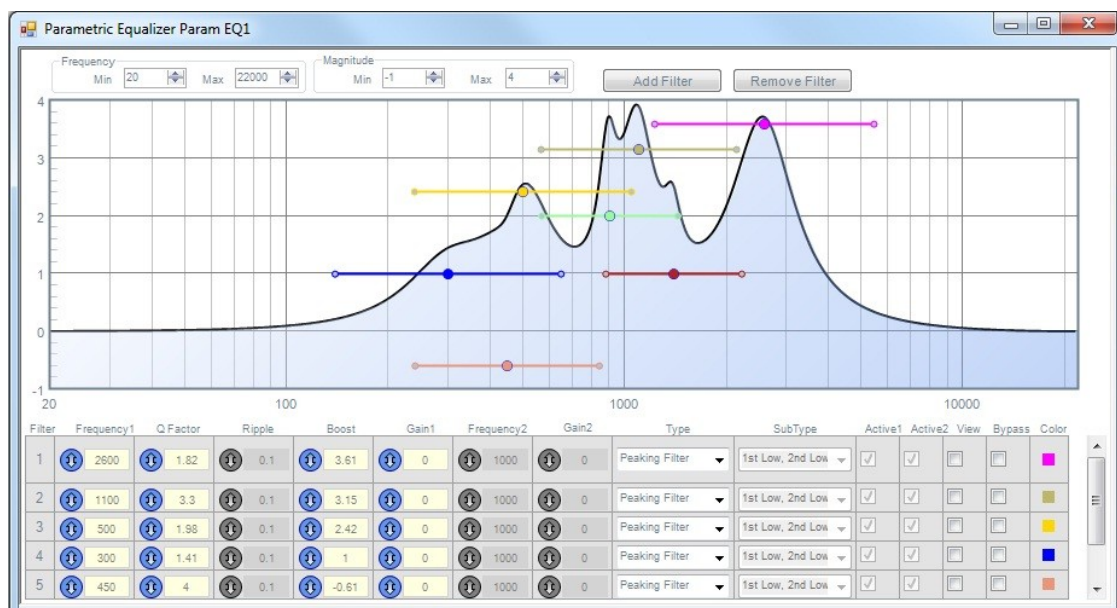
Parametrický ekvalizér a funkce ABL

V příkladu, který je uveden na obr. 63, je zobrazení ekvalizace frekvenční charakteristiky reproduktoru.

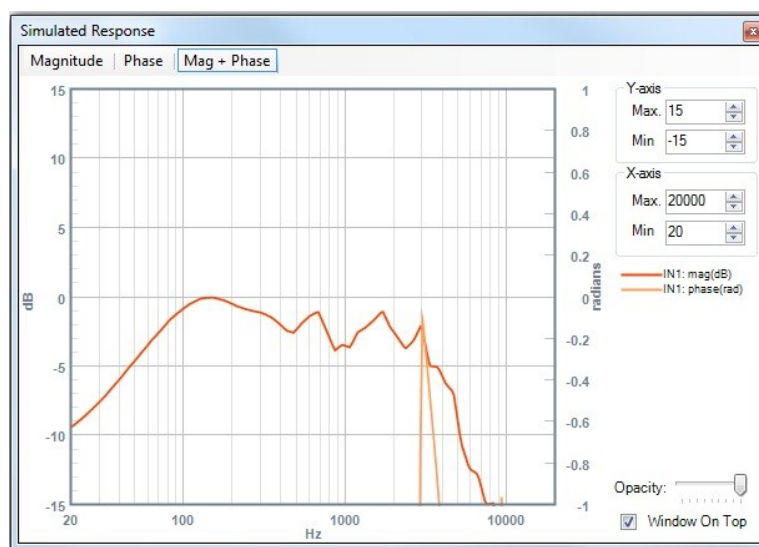
Charakteristika reproduktoru je načtena v bloku Speaker Response z textového souboru, jež je uložen ve formátu MLSSA. V sérii s blokem Speaker Response je zařazen blok parametrického ekvalizéru, jehož nastavení je zobrazeno na obr. 64. Při zobrazení přenosu pomocí Stimulus a Probe je možné měnit nastavení ekvalizace a zároveň sledovat změnu přenosu. Tímto způsobem byla vyrovnána načtená charakteristika reproduktoru, viz obr. 65, na charakteristiku zobrazenou na obr. 66.



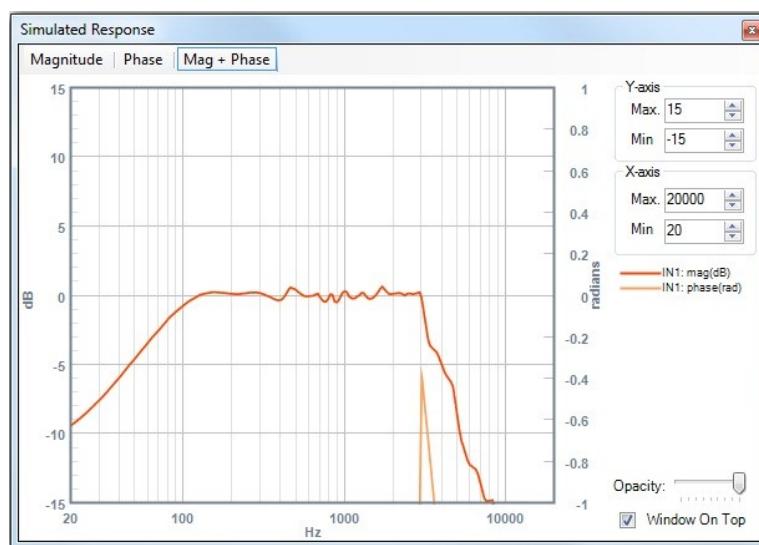
obr. 63. Příklad zapojení ekvalizéru pro vyrovnání charakteristiky reproduktoru



obr. 64. Nastavení ekvalizéru



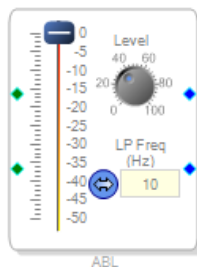
obr. 65. Charakteristika reproduktoru



obr. 66. Charakteristika reproduktoru po průchodu ekvalizérem

Z porovnání obr. 65 a obr. 66 je zřejmá ekvalizace, která napomohla vyrovnání původní charakteristiky. Pro vyrovnání bylo použito sedm parametrických ekvalizérů.

Poslední realizovanou funkcí, která je použita v této práci, je funkce ABL. Funkce je realizována blokem zobrazeným na obr. 67, který se připojí do řetězce stejným způsobem jako v předchozích příkladech.



obr. 67. Blok ABL funkce v IDE SigmaStudio

8 Reálná měření

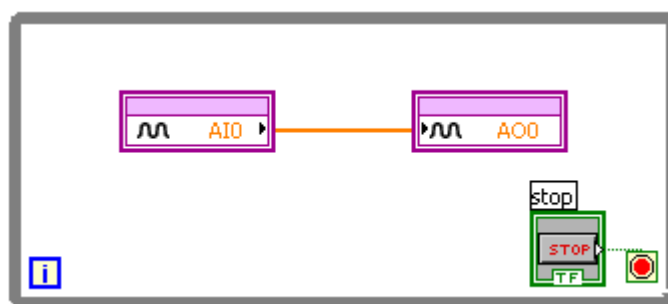
Tato kapitola se věnuje popisu a výsledkům měření. Měření bylo vyhotoveno pomocí audio analyzáru ROHDE & SCHWARZ UPV, jehož přesný popis je na [21].

Při měření byl nastaven frekvenční rozsah 20Hz až 20kHz, kdy frekvenční osa byla popsána logaritmičtí tisíci měřenými vzorky. Na signálovém generátoru přístroje byla nastavena amplituda 0,1V, která se pro měření zdála vyhovující. K přístroji byla připojena diferenciální sonda s nulovým útlumem. Před započítím měření byl zkontrolován přenos samotného přístroje, který se rovnal 1:1.

Jednotlivá měření byla uložena do souboru s příponou .LIM, který bylo možné otevřít v textovém editoru. Naměřená data byla následně importována do sešitu MS Excelu. Před samotným uložením v cílovém souboru byl průběh nejdříve v jiném sešitu znormalizován. Následně bylo vykresleno 18 grafů, které obsahují 55 měření.

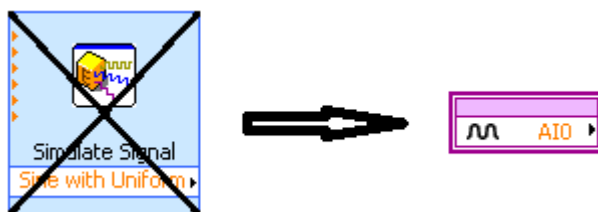
8.1 Měření úloh na kitu NI Speedy - 33

Před započítím měření vlastních úloh byl proměřen přenos samotného kitu (obr. 70), při kterém byl vstup připojen ke frekvenčnímu generátoru analyzáru a výstup byl přiveden přes měřicí sondu na vstup analyzáru. V LabVIEW DSP modulu byl vytvořen jednoduchý program, který pouze propojil jeden vstupní kanál s jedním výstupním (viz obr. 68).

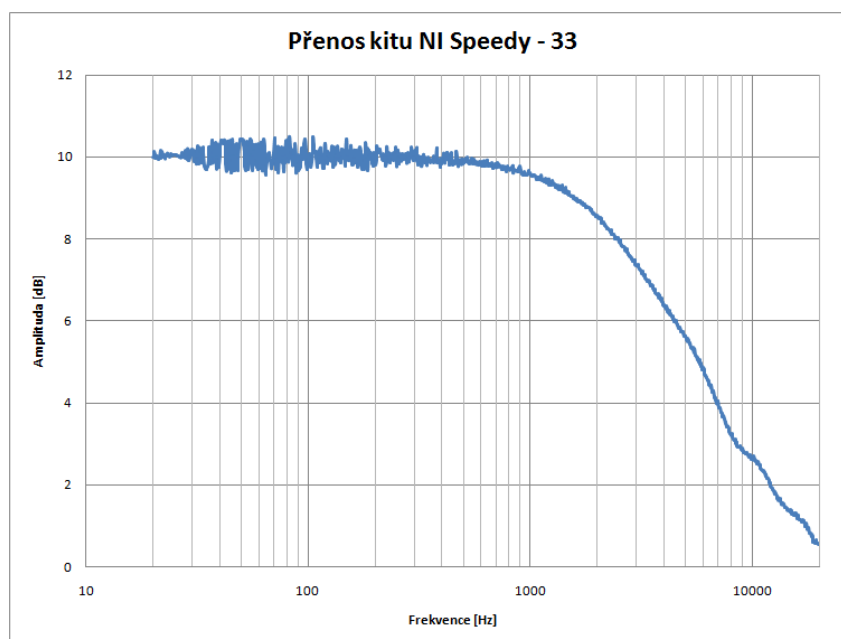


obr. 68. Algoritmus pro měření přenosu kitu

Před započítím měření jednotlivých úloh byl vždy zaměněn blok Simulate Signal za reálný analogový vstup. Analogový výstup, jehož blok je zobrazen na obr. 68, byl připojen k výstupu filtru. Oba bloky byly nastaveny na vzorkovací frekvenci 48kHz. Počet vzorků, který byl nastaven na analogovém vstupu při měření, odpovídal simulovaným úlohám.



obr. 69. Záměna bloku Simulate Signal za analogový vstup



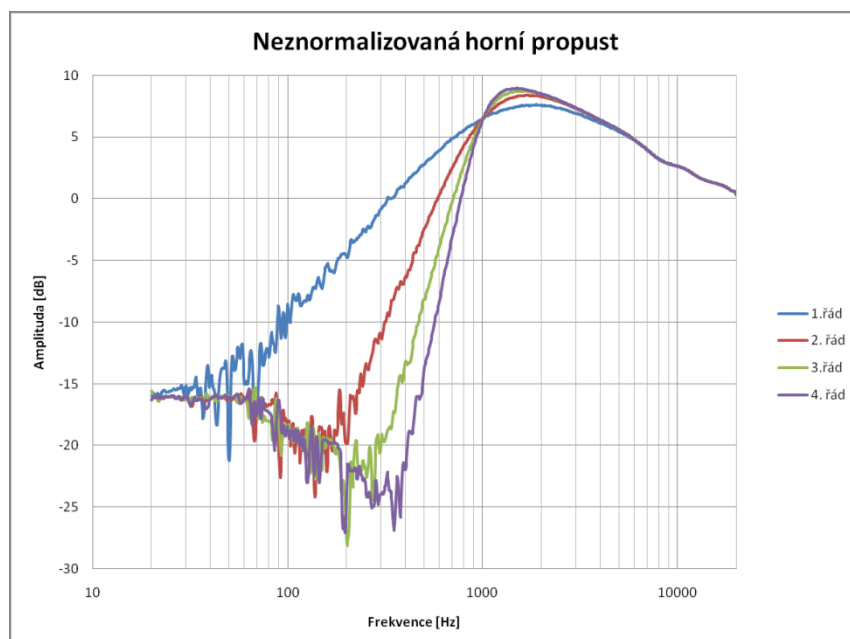
obr. 70. Přenos kitu NI Speedy – 33

Z přenosové charakteristiky kitu NI Speedy – 33 je zřejmý útlum o necelých 10 dB, jehož počátek je na frekvenci 500 až 600Hz a končí měřením ve 20kHz. Také je patrný vstupní šum, který je obsažen ve frekvencích od 25 – 600Hz, jehož maximální amplituda je 0,5dB.

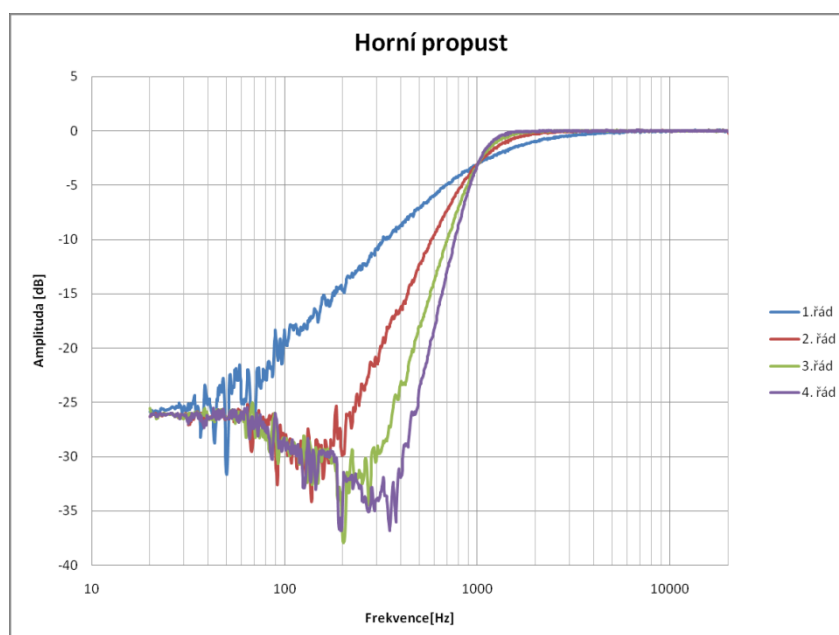
Z důvodu správného zobrazení naměřených charakteristik jsou výsledné průběhy normalizovány naměřeným přenosem kitu. Normalizace spočívá v odečtení přenosové charakteristiky kitu od naměřené charakteristiky konkrétní úlohy.

Na následujících dvou grafech (obr. 71 a obr. 72) je vykreslen naměřený průběh a znormalizovaný průběh horních propustí. Z výsledků je patrné, že tato normalizace je pro správné zobrazení nutná. V případě vykreslení neznormalizovaných charakteristik by se v případě HP výsledek podobal pásmové propusti. Po znormalizování charakteristik již většina průběhů odpovídá teoretickým předpokladům.

Ve všech případech měření dolních a horních propustí byla zlomová frekvence nastavena na 1kHz. U měření pásmových propustí byly zlomové frekvence voleny na 100Hz a 2kHz.

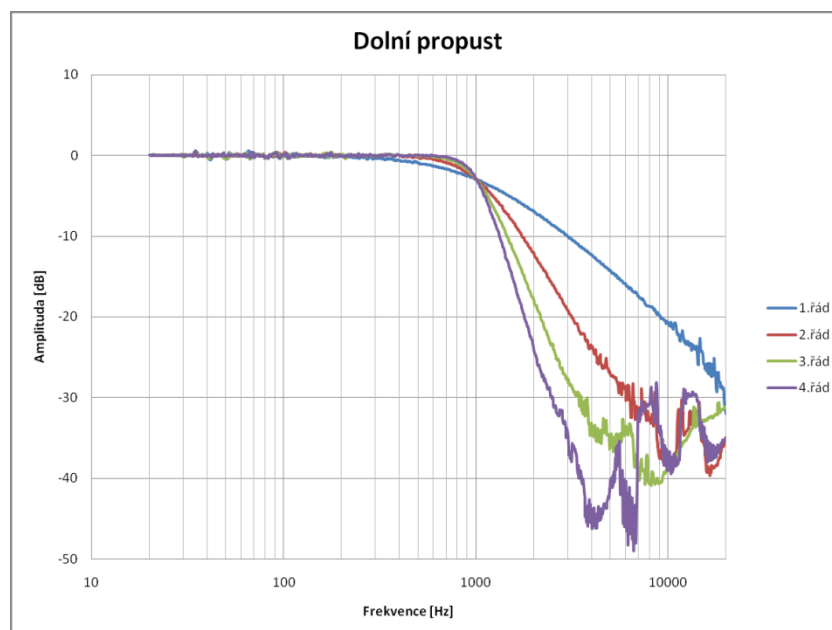


obr. 71. Neznormalizovaná horní propust 1. – 4. řádu



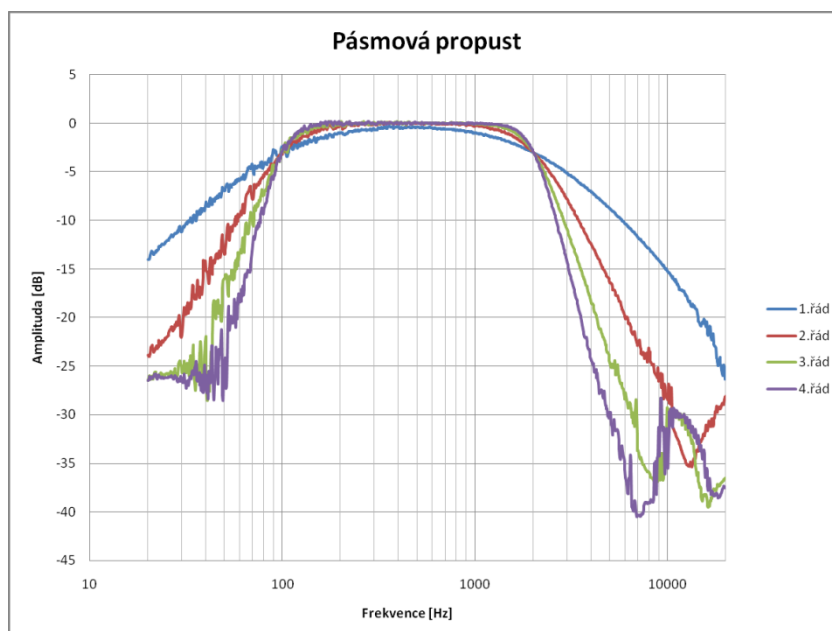
obr. 72. Horní propust 1. – 4. řádu

Z grafu na obr. 72 je patrná správná funkčnost filtru s nastavením zlomové frekvence 1kHz. Zákmity charakteristik v nepropustném pásmu jsou pravděpodobně způsobeny rušením.



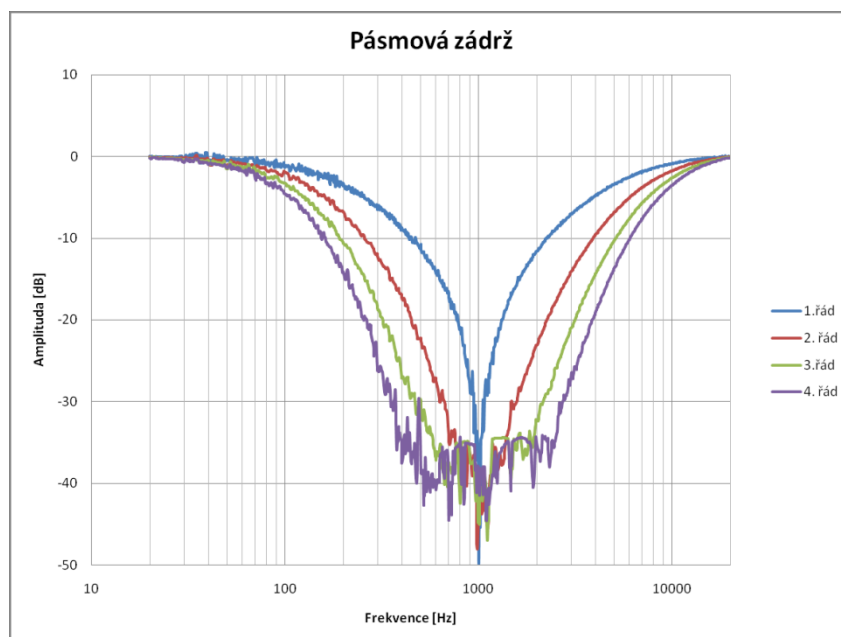
obr. 73. Dolní propust 1. – 4. řádu

Dolní propust s nastavením zlomové frekvence 1kHz a řádem filtru 1. – 4. pracuje dle teoretických předpokladů. Jednotlivé překmity v nepropustném pásmu by odpovídaly Gibbsovu jevu [22]. Ten popisuje zákmity Furierových řad, které se vyskytují na diskontinuitách diskrétního systému.



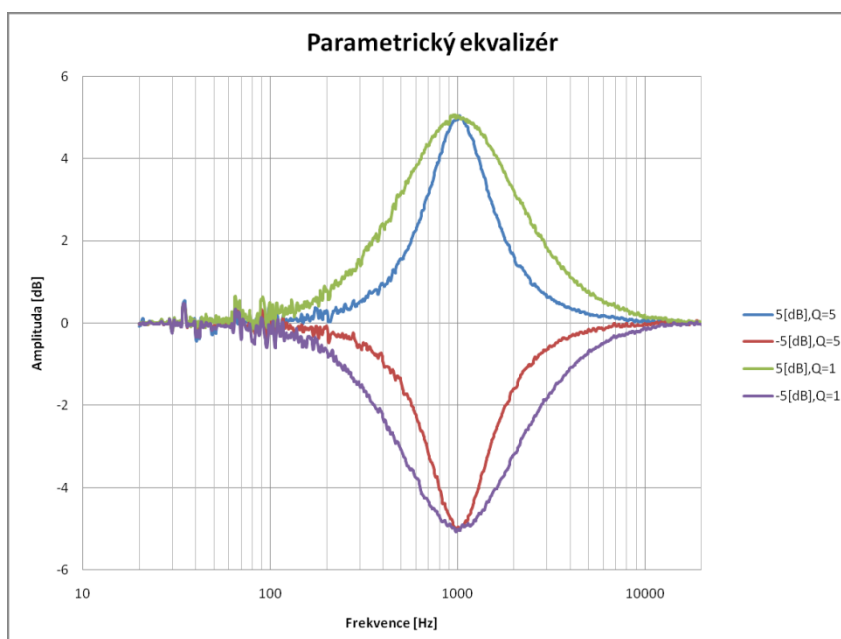
obr. 74. Pásmová propust 1. – 4 řádu.

Pásmová propust 1. – 4. řádu s nastavením dělicích kmitočtů 100Hz až 2kHz opět pracuje dle teoretických předpokladů. Zákmity charakteristik jsou pravděpodobně způsobeny Gibbsovým jevem a rušením.



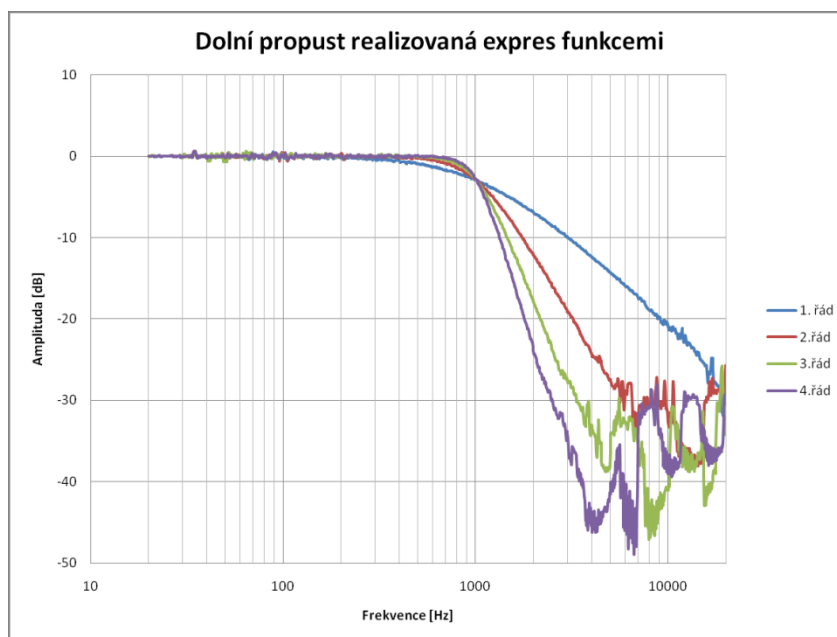
obr. 75. Pásmová zadrž 1. – 4. řádu

Na grafu obr. 75 je znázorněna pásmová zadrž 1. – 4. řádu. Z obrázku je patrný spád křivky pro jednotlivé řády. Naměřené průběhy také odpovídají již zmíněné chybě (podkapitola 4.1.2) nastavení šířky pásma, kdy 4. řád má nastavenou hodnotu 4 krát vyšší než řád první.



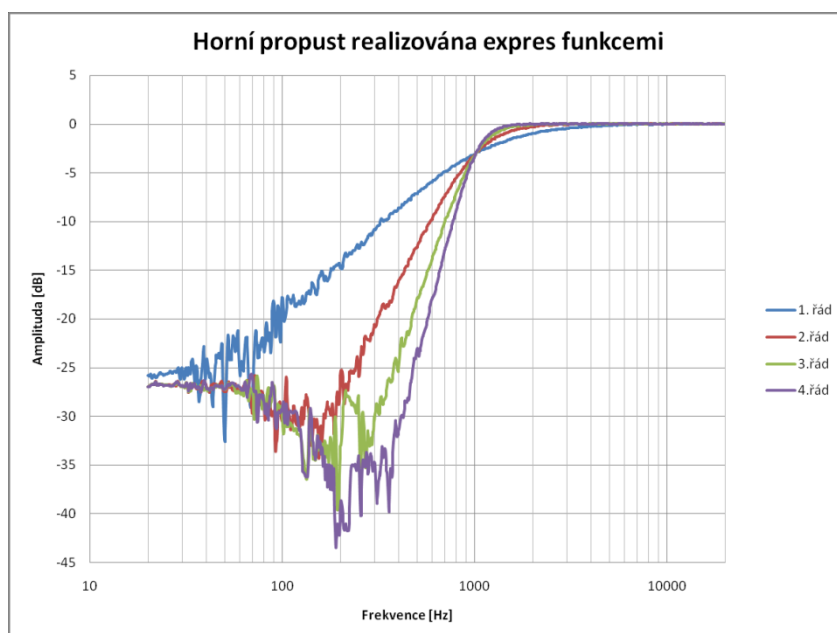
obr. 76. Parametrický ekvalizér

Na obr. 76 jsou znázorněny čtyři průběhy parametrického ekvalizéru pro nastavení $Q=1$; $Q=5$, zesílení=5; zesílení=-5. Parametrický ekvalizér se opět chová dle ověřených výsledků a teoretických předpokladů.

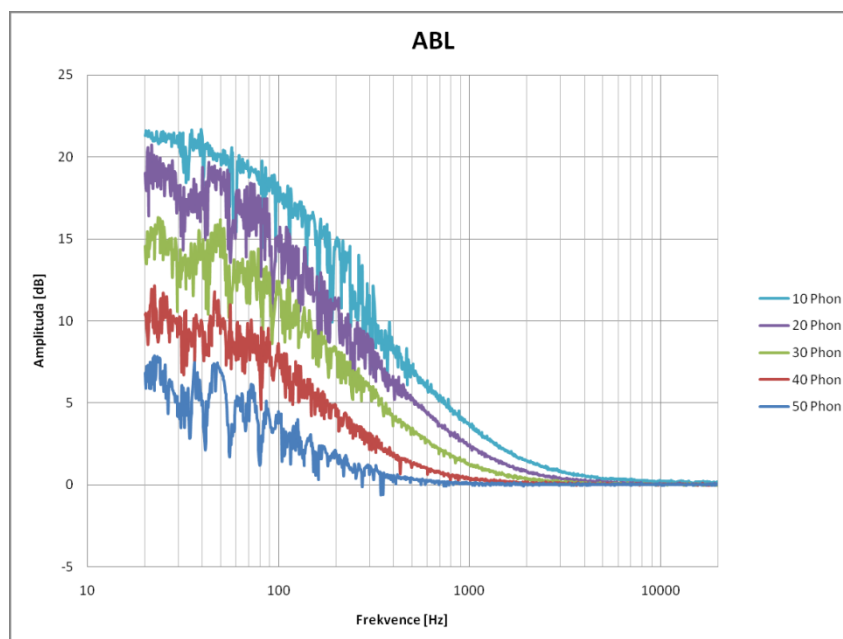


obr. 77. Dolní propust 1. – 4. řádu realizovaná expres funkcemi

Z průběhů zobrazených na obr. 77 a obr. 73 je patrná podobnost, která srovnává filtraci navržených bikvadratických filtrů a filtrů užívající expresních funkcí. Opět se potvrzuje správnost návrhu a korelace výsledků s teoretickými předpoklady. Průběhy se liší pouze v zákmitech charakteristiky v nepropustném pásmu a rušením znatelným na nižší kmitočet. Analogické srovnání horní propusti je na obr. 78 a obr. 72.

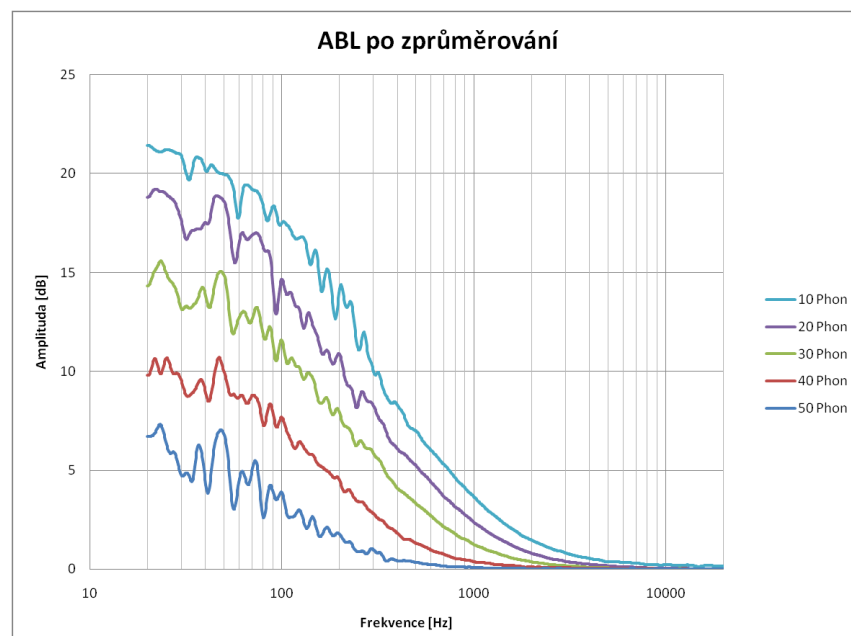


obr. 78. Horní propust 1. – 4. řádu realizovaná expres funkcemi



obr. 79. Funkce ABL

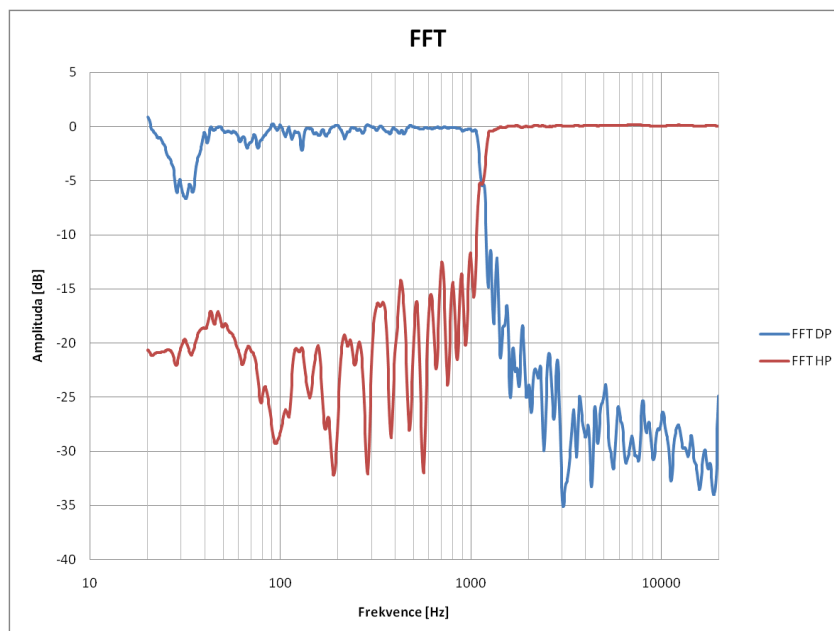
Z výsledku měření, které jsou patrné na obr. 79, je zřejmá funkčnost nastavení zesílení jednotlivých úrovní ve phonech. Důvod zákmitů charakteristik je fázový posun mezi jednotlivými PP, jež je podrobněji popsán v podkapitole 6.2. Tato chyba byla v uvedeném kódu odstraněna.



obr. 80. Funkce ABL po vyhlazení klouzavým průměrem

Pro lepší zřetelnost byly původní průběhy z obr. 79 vyhlazeny klouzavým průměrem [23] počítaným ze sedmi, pěti, a tří vzorků. Zvlnění charakteristik jednotlivých nastavení má

obdobný průběh, což by mohlo odpovídat jednotlivým fázovým posunům, které se budou uplatňovat se stejnou mírou. Například v oblasti 40 – 55Hz je patrné převýšení, jež se zřetelně objevuje na křivkách označených 20 – 50 phonů.



obr. 81. Filtrace pomocí FFT

Průběhy na obr. 81 zobrazují dolní a horní propust realizovanou pomocí FFT filtrace, jež byly vyhlazeny metodou klouzavých průměrů jako v předchozím příkladě. Z měření je zřejmé, že ideální matematický model nelze použít v praxi bez úprav. Pro správnou funkci by bylo nutné snížit náběh přechodu mezi propustným a nepropustným pásmem. To je běžně realizováno řádem filtru. V tomto případě je řád vlastně nekonečný a tak bude opět docházet ke Gibbsovu jevu, který se projevil v zakmitaném průběhu obou naměřených výsledků.

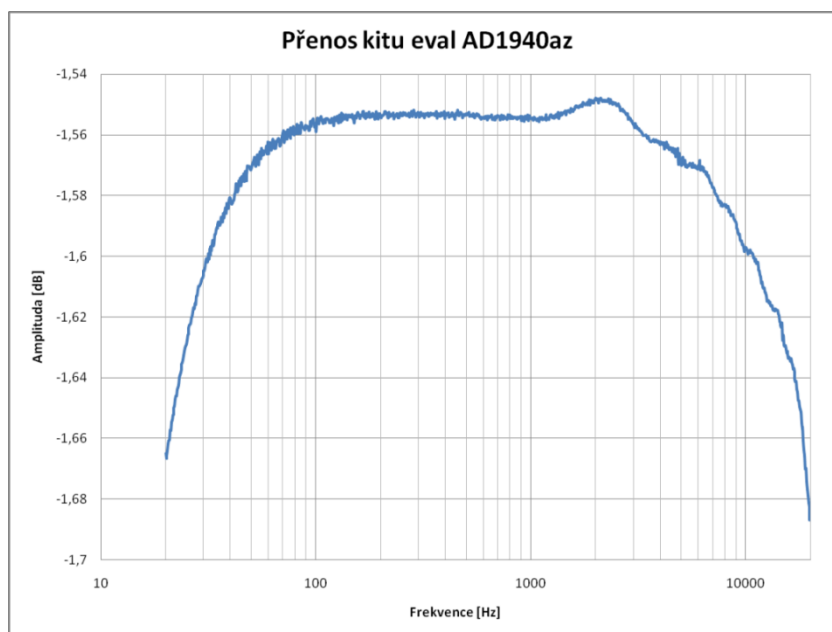
8.2 Měření úloh na kitu EVAL AD1940AZ

Pro srovnání výsledků kitu NI Speedy – 33 byla změřena sada obdobných úloh na vývojovém kitu EVAL AD1940AZ. Postup měření byl analogický jako u kitu NI Speedy – 33, avšak pro demonstraci byly proměřeny pouze filtry dolní, horní, pásmové propusti 2. až 4. řádu, sedmi pásmový parametrický ekvalizér a funkce ABL.

Všechny naměřené výsledky odpovídají simulovaným průběhům ve vývojovém prostředí SigmaStudio bez jakýchkoli významných překmitů.

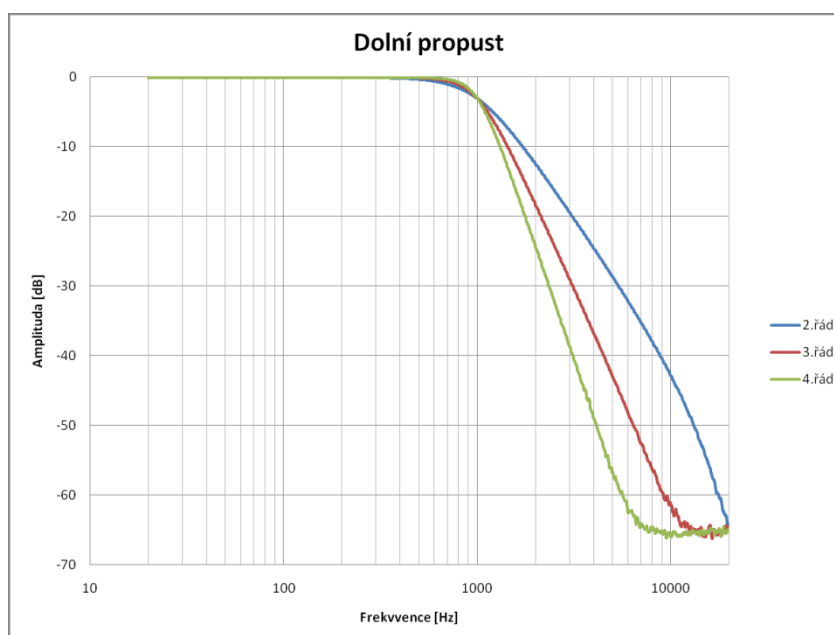
Před započítím měření byl nejdříve změřen přenos kitu EVAL AD1940AZ, který sloužil pro normalizaci výsledků jednotlivých úloh.

Nastavení audio analyzáru ROHDE & SCHWARZ UPV a zlomové frekvence jednotlivých filtrů bylo shodné s měřením, jež bylo použito u kitu NI Speedy -33.



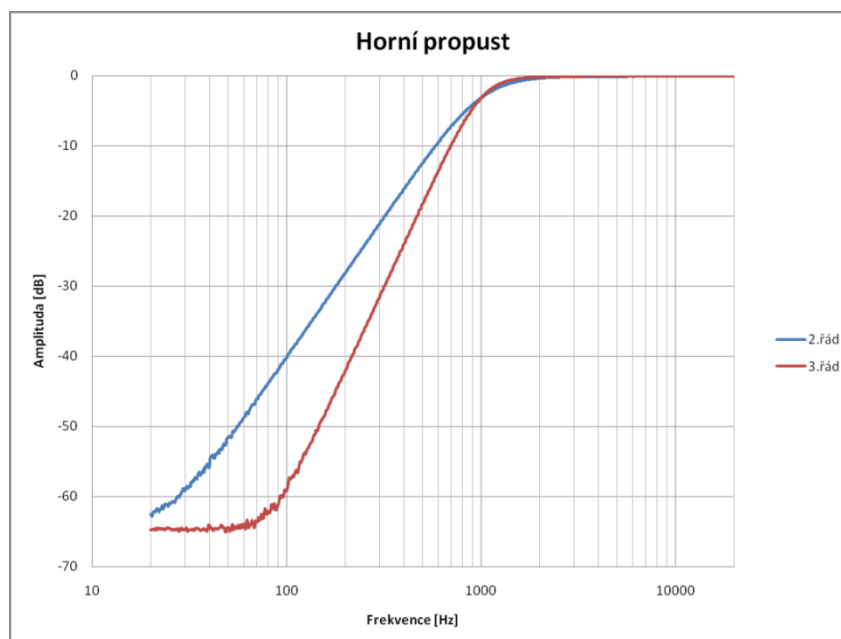
obr. 82. Přenos kitu EVAL AD1940AZ

Z přenosu kitu, který je na obr. 82, je patrný útlum přibližně -1,55dB. V oblasti kmitočtů 20 – 100Hz a 3 – 20kHz je patrný útlum až 1,685dB. Tento rozdíl je však nepatrný, a proto se dá považovat přenos kitu za téměř ideální.



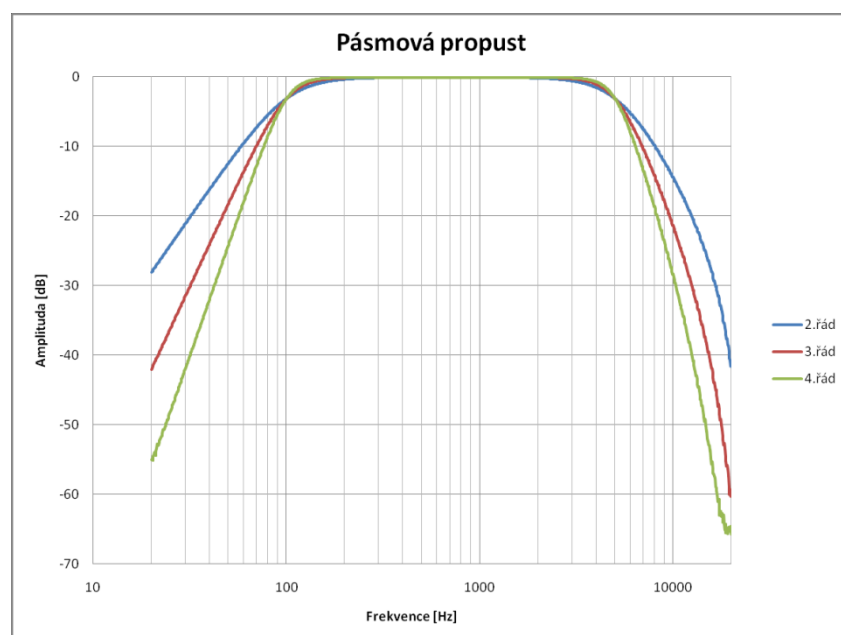
obr. 83. Dolní propust 2. – 4. řádu

Z naměřených průběhů dolní propusti je zřejmá správná funkčnost jednotlivých řádu filtrů. Naměřené charakteristiky odpovídají simulovaným návrhům, avšak při simulaci průběh končí vždy v zobrazované oblasti. V případě reálného měření je docílen útlum na -65dB, kde se již charakteristika ustálí.



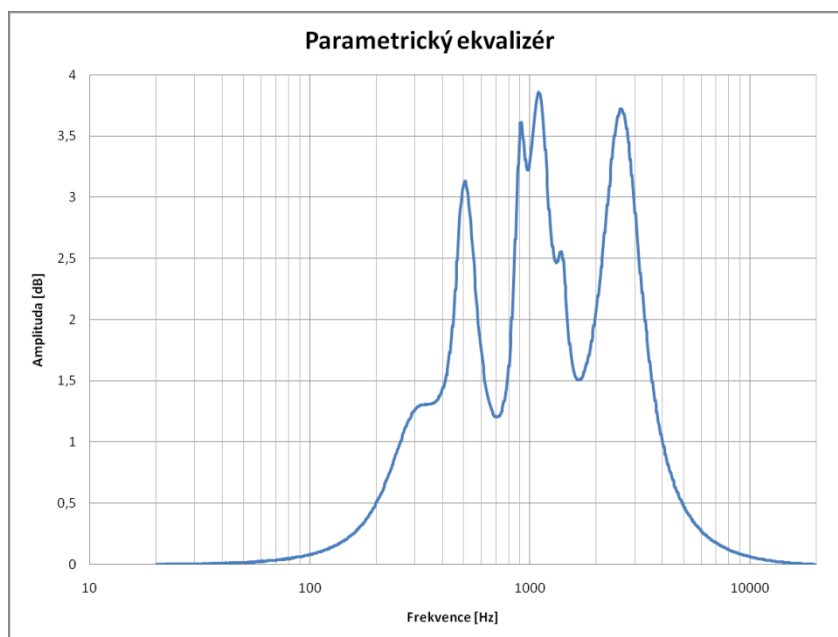
obr. 84. Horní propust 2. a 3. řádu

Z naměřených charakteristik na obr. 84 je opět zřejmá správná funkčnost, jež odpovídá simulaci a teoretickým předpokladům. Chybějící průběh horní propusti 4. řádu byl při měření omylem přemazán jiným výsledkem.



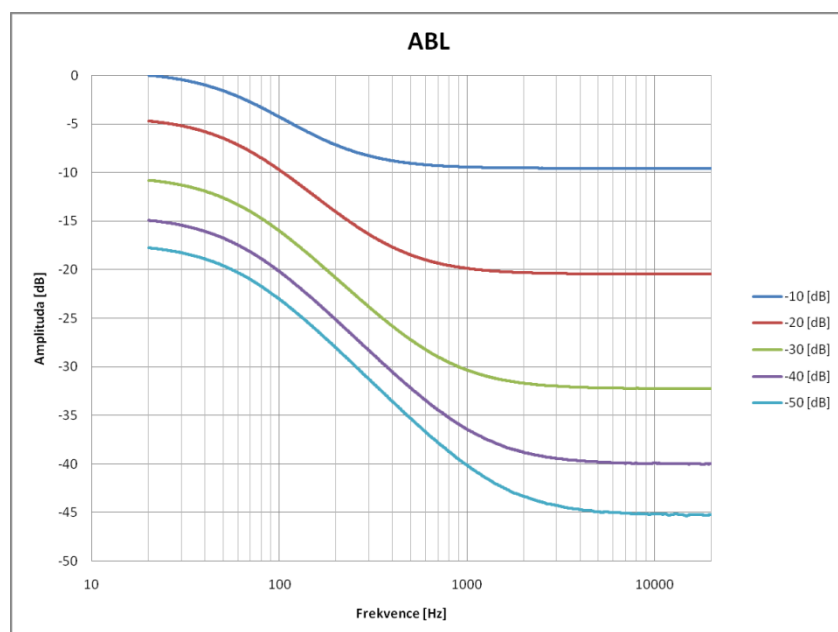
obr. 85. Pásmová propust

Pásmové propusti, jež jsou v grafu na obr. 85, opět odpovídají simulaci a teoretickým předpokladům. V tomto případě byl nastaven dělicí kmitočet na 100 Hz a 5 kHz.



obr. 86. Ekvalizér

Z porovnání simulace, jež je na obr. 64, a měřeného výsledku na obr. 86 je na první pohled zřejmá totožnost. Ekvalizér se chová ideálně bez jakýchkoli znatelných překmitů.



obr. 87. ABL funkce realizovaná na kitu AD1940az

Na obr. 87 jsou zobrazeny průběhy naměřených charakteristik funkce ABL v SigmaStudios. V tomto případě jsou jednotlivá zesílení basových kmitočtů svázána s potenciometrem celkového zesílení. Proto jsou tyto průběhy vzájemně posunuty o nastavený útlum, který je v dB. Platí zde tedy předpoklad, že čím bude nižší zesílení, tím více se bude zesilovat basové spektrum signálu, což odpovídá požadavkům funkce ABL.

9 Závěr

Cílem diplomové práce byl průzkum vlastností kitu NI Speedy -33 a návrh filtrů prvního a vyšších řádů pro horní, dolní a pásmovou propust, pásmovou zádrž a parametrický ekvalizér, které jsou určeny pro implementaci v reproduktorových soustavách. Dalším úkolem bylo vytvoření funkce ABL a reálné měření všech navržených filtrů.

Největším omezením kitu NI Speedy – 33 je nedostatečný počet výstupních kanálů, které jsou pouze dva. Proto je možné na kitu zpracovávat pouze dvě frekvenční pásma výstupního signálu. Kit spolupracuje pouze s LabVIEW DSP modulem, který je dostupný pro LabVIEW do verze 8.4. Kit lze programovat z grafického vývojového prostředí LabVIEW. To umožňuje snadnou a rychlou implementaci navrženého algoritmu. Proto je kit vhodný pro demonstraci typických úloh zpracovávaných pomocí DSP. Výrobce kitu NI Speedy – 33 neposkytuje schéma zapojení, pouze popis konektorů. Odladěné programy tak nelze přenést na jiný hardware, a proto není možné vytvořit konečný výrobek bez použití vlastního kitu.

Všechny naprogramované úlohy byly před měřením simulovány přímo na HW kitu. Jako zdroj signálu byl použit blok Simulate Signal, který byl nastaven s požadovanými parametry pro danou úlohu. Po odsimulování úloh, jež se jevily, že pracují korektně, bylo přistoupeno k měření. Pro měření byl použit audio analyzátor ROHDE & SCHWARZ UPV, který poskytla firma Bang and Olufsen. První měření spočívalo v proměření přenosu samotného kitu NI Speedy – 33. Naměřená charakteristika odhalila špatné elektrické vlastnosti kitu, které se projevovaly rušením v oblasti kmitočtů mezi 20 až 500Hz s amplitudou přibližně 0,5dB. Pro správné zobrazení naměřených charakteristik bylo nutné normovat jednotlivé výsledky naměřeným přenosem kitu. Výsledky normovaných charakteristik byly přeneseny do grafů, ze kterých je zřejmá správná funkce většiny filtrů, jež odpovídá teoretickým předpokladům.

Měřením byly odhaleny dvě chyby v návrhu algoritmů. První chyba vznikla při měření pásmové zádrže, kdy byla v naprogramovaném algoritmu špatně zadána šířka pásma. Chyba se projevila tak, že zadaná šířka pásma byla vynásobena aktuálním řádem filtru. Což znamená, že v případě 4. řádu byla šířka pásma 4x větší než u řádu prvního. Kód uvedený v diplomové práci je již opraven. Druhá chyba byla objevena v návrhu funkce ABL, kde bylo nesprávně použito pásmových propustí, které měly funkci pásmového ekvalizéru. Použití této metody ekvalizéru publikoval profesor Karam z Arizonské university USA. Tato metoda je chybná, jelikož jednotlivé pásmové propusti mají jiný fázový posuv. Z tohoto důvodu je výstupní signál, který je tvořen součtem jednotlivých pásem ekvalizéru značně zakmitaný. Z důvodů zákmitů jednotlivých charakteristik byla použita metoda klouzavých průměrů, která jednotlivé křivky naměřených průběhů vyhladila.

Pro zajímavost byl vytvořen příklad filtru dolní a horní propusti ve frekvenčním spektru. Algoritmus filtru byl založen na nulování určitých amplitud ve frekvenčním spektru. Při simulaci na HW kitu se aplikace chovala dle teoretických předpokladů. Při měření však docházelo k četným zákmitům charakteristiky. Zákmity pravděpodobně způsoboval příliš strmý přechod mezi propustným a nepropustným pásmem. Pro lepší výsledek byla charakteristika opět vyhlazena metodou klouzavých průměrů. Z důvodů nedostatku paměti HW kitu NI Speedy – 33 bylo možné použít při FFT filtraci příliš nízký počet vzorků. Jednotlivé vzorky popisují amplitudu na dané frekvenci. Počet vzorků, který byl v úloze použit, měl hodnotu 512. To znamená, že krok frekvence osy x, při zvolené vzorkovací frekvenci 44,1kHz byl 86,13Hz. Tento krok je pro filtraci v oblasti basových kmitočtů příliš hrubý. Některé transformace popisují osu x logaritmicky, což řeší danou problematiku. Tuto možnost však blok FFT

v LabVIEW DSP modulu neumožňuje. Z těchto důvodů je filtrace ve frekvenčním spektru na kitu NI Speedy – 33 nepoužitelná.

Nad rámec této práce byly vypracovány úlohy dolní, horní a pásmové propusti, parametrický ekvalizér a funkce ABL na kitu EVAL AD1940AZ. Kit EVAL AD1940AZ je taktéž programovatelný přes grafické vývojové prostředí - SigmaStudio. V tomto případě byly jednotlivé úlohy simulovány pouze ve vývojovém prostředí. Měření proběhlo obdobně, jak bylo popsáno u kitu NI Speedy – 33. Opět byla změřena přenosová charakteristika kitu, která byla téměř ideální. Kit EVAL AD1940AZ vykazuje téměř lineární charakteristiku s poklesem amplitudy o 1,55dB. I přesto byly naměřené průběhy touto křivkou normovány.

Výsledky měření odpovídají teoretickým a simulovaným předpokladům a jednotlivé průběhy jsou bez jakýchkoli znatelných zákrmitů. Například změřená charakteristika parametrického ekvalizéru je identická se simulovanou.

Na základě porovnání průběhů přenosových charakteristik kitů je zřejmé, že kit NI Speedy – 33 je naprosto nevhodný pro použití v digitálním zpracování audio signálu. Naměřený pokles amplitudy kitu NI Speedy - 33 o -10dB je v audio technice nepoužitelný. Kvalitní reproduktorová soustava by neměla překročit hranici (útlumu nebo zdvihu) $\pm 3\text{dB}$ v celém slyšitelném spektru. Kit je však určen pro demonstraci použití DSP procesoru v různých aplikacích, které plně dostačuje. Vývojový kit EVAL AD1940AZ je určen pro zpracování audio signálu, o čemž vypovídají naměřené charakteristiky. Výrobce kitu EVAL AD1940AZ na svých stránkách uvádí kompletní schéma zapojení. Na základě publikovaného schématu bylo vyhotoveno nové schéma, podle kterého bude navrhuta deska plošných spojů. V budoucnu bude naprogramován mikrokontrolér pro řízení komunikační sběrnice (SPI nebo I2C), který bude ovládat parametry DSP Sigma. Tím bude vytvořen základ pro vývoj aplikací určených pro koncového uživatele.

10 Použité zdroje

- [1] SKALICKÝ, P. *Aplikace signálových procesorů*. 1st ed. Praha: ČVUT, 2005. 136 p. ISBN 80-01-02647-7.
- [2] SMITH, S. W. *The Scientist & Engineer's Guide to Digital Signal Processing*. 1stth ed. California: California Technical Pub, 1997. 626 p. ISBN -10 0966017633.
- [3] PROAKIS, J. G., MANOLAKIS, D. G. *DIGITAL SIGNAL PROCESING: Principles, Algorithms, and Applications*. 2nd ed. New York: Macmillan Publishing Company, 1992. 969 p. ISBN 0-02-236815-x.
- [4] ŠEBESTA, J. *Mikroprocesorová technika: skriptum*. Brno: VUT, 2002.
- [5] SMÉKAL, Z., SYSEL, P. *Signálové procesory*. 1st ed. Praha: Sdělovací technika, 2006. 281 p. ISBN 80-86645-08-8.
- [6] VLACH, J., HAVLÍČEK, J., VLACH, M. (ed.). *Začínáme s LabVIEW*. 1st ed. Praha: BEN - technická literatura, 2008. 284 p. ISBN 978-80-7300-245-9.
- [7] DAVÍDEK, V., LAIPERT, M., VLČEK, M. *Analogové a číslicové filtry*. 2nd ed. Praha: ČVUT, 1999. 345 p. ISBN 80-01-03026-1.
- [8] SOVKA, P., POLLÁK, P. *Vybrané metody číslicového zpracování signálů*. 1st ed. Praha: ČVUT, 2001. 255 p. ISBN 80-01-02821-6.
- [9] DAVÍDEK, V. *Implementace algoritmů číslicového zpracování signálů v reálném čase*. 1st ed. Praha: ČVUT, 2004. 161 p. ISBN 80-01-03114-4.
- [10] UHLÍŘ, J., SOVKA, P., ČMEJLA, R. *Úvod do číslicového zpracování signálů*. 1st ed. Praha: ČVUT, 2003. 161 p.
- [11] ŽÍDEK, J. *Grafické programování ve vývojovém prostředí LabVIEW*. Ostrava: ČVUT, 2002. 215 p.
- [12] JAN, J. *Číslicová filtrace, analýza a restaurace signálů*. 2nd ed. Praha: ČVUT, 2002. 417 p. ISBN 80-214-1558-4.
- [13] ORSAK, G., et al. *Engineering Our Digital Future*. 1st ed. Upper Saddle River: Prentice Hall PTR, 2004. 483 p. ISBN 0-13-143643-0.
- [14] NASSER, K., NAMJIN, K., et al. *Digital Signal Processing System - Level Design*. 1st ed. Burlington, 2005. 287 p. ISBN 0-7506-7914-x.
- [15] TRAVIS, J. *LabVIEW FOR EVERYONE*. 2nd ed. Upper Saddle River: Prentice Hall PTR, 2002. 589 p. ISBN 0-13-065096-x.
- [16] VLACHÝ, V. *Praxe Zvukové Techniky*. 2nd ed. Praha: Muzikus, 2000. 257 p. ISBN 80-86253-05-8.
- [17] LES, T., et al. *Practical Analog and Digital Filter Design*. 1st ed. Ohio, 2004. 265 p.
- [18] VLACHÝ, V. *Reproduktory a reprosoustavy*. 1st ed. Orlová: Dexon, 2003. 205 p.
- [19] Tackett, J. ISO 226 Equal-Loudness-Level Contour Signal, 2005. MATLAB CENTRAL. <http://www.mathworks.com/matlabcentral/fileexchange/7028-iso-226-equal-loudness-level-contour-signal> (accessed April 10, 2012).
- [20] Karam, L. *Introduction to Signal Processing in Engineering Laboratory Manual* [online]; Arizona State University: USA, 2007. https://trac.rhaptos.org/trac/rhaptos/raw-attachment/ticket/4260/eluther__Karam_Lab_Manual_edited.doc (accessed April 05, 2012).
- [21] UPV_dat_sw_en, 2011. ROHDE & SCHWARZ. http://www2.rohde-schwarz.com/file_16563/UPV_dat_sw_en.pdf (accessed April 29, 2012).

- [22] Máša, P. *Ortogonalita, Koeficienty Furierove řady, Gibbsův jev*, 2006. Výpočetní a informační centrum ČVUT.
www.civ.cvut.cz/info/download_prednasky.php?name=eo2_01b.pdf (accessed April 29, 2001).
- [23] V. Rogaliewitz, *Stochastické procesy (Analýza časových řad)*, Praha, 1993
- [24] TMS320vc33. *Texas Instruments*. <http://www.ti.com/lit/ds/symlink/tms320vc33.pdf> (accessed March 20, 2012).
- [25] DSP56301. Freescale.
http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=DSP56301 (accessed April 20, 2012).
- [26] ADSP21020. Datasheet. <http://www.datasheetdir.com/ADSP-21020+Embedded-Processing-DSP> (accessed April 20, 2012).
- [27] DSP56300. Ecse. <http://www.ecse.rpi.edu/courses/CStudio/dsp56303/56300fm.pdf> (accessed April 20, 2012).
- [28] Od mikrořadičů k digitalním signálovým procesorům DSP. Root.
<http://www.root.cz/clanky/od-mikroradicu-k-digitalnim-signalovym-procesorum-dsp/> (accessed March 13, 2012).
- [29] Digitální signálové procesory v průmyslu. Automatizace.
<http://www.automatizace.cz/article.php?a=336> (accessed March 17, 2012).
- [30] IIR filter function. IIR filter function.
http://software.intel.com/sites/products/documentation/hpc/compilerpro/en-us/cpp/lin/ipp/ipp/ipp_ch6/ch6_iir_filter_functions.html (accessed Jan 01, 2012).
- [31] Smith, J. *Introduction to Digital Filters*. <https://ccrma.stanford.edu/~jos/fp/> (accessed Jan 01, 2012).
- [32] Biquad design. <http://www.arachnoid.com/BiQuadDesigner/index.html> (accessed Jan 01, 2012).
- [33] Bristow, R. EQ - coefficients. <http://www.musicdsp.org/files/EQ-Coefficients.pdf> (accessed April 09, 2012).
- [34] Low pass filter. <http://www.kwon3d.com/theory/filtering/lpass.html> (accessed March 09, 2012).
- [35] Vassilevsky, V. *A Practical Guide for Digital IIR Audio Filters*, 2003. AbVolt.
<http://www.abvolt.com/research/publications2.htm> (accessed April 09, 2012).
- [36] ŠALOUN, P. *Programovací jazyk C++*. 1st ed. Praha: Neocortex, 2005. 252 p. ISBN 80-86330-18-4.
- [37] HEROUT, P. *Učebnice jazyka C*. 4th ed. České Budějovice: Kopp, 2007. 271 p. ISBN 80- 7232-220-6.
- [38] NI Speedy - 33, 2008. National Instruments.
<http://www.ni.com/pdf/manuals/371577d.pdf>.
- [39] Graphical Dataflow Programming: LabVIEW and Other Tools. Kuro5hin.
<http://www.kuro5hin.org/story/2003/2/3/1688/50511> (accessed March 04, 2012).
- [40] ISO 226. *Akustika - Normované křivky stejné hlasitosti*. Praha: ČESKÁ TECHNICKÁ NORMA, 2005. 20 p.

11 Přílohy

- I. Navržené elektrotechnické schéma pro procesory ADAU144x
- II. Naměřené výsledky uložené v sešitu MS Excel
- III. Algoritmy úloh pro kit NI Speedy - 33